Enabling In-Network Caching in Traditional IP Networks: Selective Router Upgrades and Cooperative Cache Strategies

Jiangping Han, *Member, IEEE,* Kaiping Xue, *Senior Member, IEEE,* Jian Li, *Member, IEEE,* Jing Zhang, Zixuan Huang, David S.L. Wei, *Senior Member, IEEE*

***Abstract*—Enabling in-network caching in a traditional IP network by progressively adding cache-enabled nodes into the network can provide a variety of advantages, such as efficient content distribution and improved network resource utilization. However, to maximize the performance, two crucial issues have to be addressed: 1) due to a limited budget, only selective nodes could be upgraded, and thus which nodes should be chosen is of importance and 2) how to decide the corresponding cache strategy in an upgraded hybrid network is also important. In this paper, we first formulate a problem of Selective Router Upgrade (SRU) aiming at selecting nodes to be upgraded such that both average access delay and hit ratio are optimized for the best experience. We prove that SRU is an NP-hard problem, and then propose a $(1-1/e)$-approximation algorithm for it. Based on the model of SRU, we further propose Local Replacement (LR) and Neighbor Cooperative Caching (NCC) strategies to provide caching service in an upgraded hybrid network. Specifically, LR considers the cache behavior in a local router for both low latency and high hit ratio. NCC allows routers to obtain contents from their neighbor nodes and further decides the caching replacement policy based on neighbor nodes' caching capabilities. In a word, LR and NCC help expand the overall caching space and improve the content delivery efficiency by exploring the geometrical vicinity of nodes. Extensive simulation results show that our schemes can significantly improve the network in terms of access delay and hit ratio.**

***Index Terms*—In-network caching, network upgrade, neighbor cooperative cache.**

I. Introduction

The Internet business model is changing from end-to-end communication to content distribution and acquisition, since it is indicated in Cisco's report [1] that network traffic will be dominated by analog data. Meanwhile, with the advances in hardware technology, the computing power and storage capacity of network nodes are continuously strengthened, which enables them to do more than simple routing. As such, a new trend in network design is to enable a network to have in-network caching [2], [3], which takes advantage of the caching capabilities of some intermediate routers to cache copies of the frequently used contents. The use of in-network-caching benefits various of applications such as IoT [4]–[6] and video streaming [7]. There have been a lot of research efforts dedicated to investigate and develop such network architectures [8]–[10]. Although their implementations are different, they all use an in-network caching-based network model. The routers in the network can cache multiple copies of those frequently accessed contents such that the consumers can access the contents from a closer place.

However, due to backward compatibility concerns and limited budget constraints, it is often impractical to completely overhaul a traditional IP network with a completely new architecture. To address this challenge and harness the performance benefits of in-network caching in traditional IP networks, various architectures have been proposed [11]. These architectures, such as CONET [12], CAIP [13], and hICN [14], enable selective cache-enablement of specific network nodes using newly defined IP options or IPv6 extension headers, while keeping other nodes unchanged. By allowing these selected nodes to support both in-network caching-based routing and IP-based routing, they facilitate the progressive integration of in-network caching capabilities. This approach provides a viable pathway for enabling in-network caching while ensuring compatibility with existing infrastructure.

Enabling in-network caching in the traditional IP network faces some new problems and challenges. In this work, we mainly focus on solving the following two problems. **First, depending on the budget, making the right decision to select a subset of routers from the set of traditional ones to be upgraded is of importance.** As shown in Fig. 1, for progressively transforming a traditional network to be fully cache-enabled, some network nodes can be upgraded to cache-enabled ones in each upgrade. Since the budget is limited, it is necessary to provide an effective solution to ensure maximum benefit in the upgrading process. **Secondly, the cache strategy and neighbor cooperation in an upgraded hybrid network also affect access delay and hit ratio significantly.** When some of the routers are upgraded to enable in-network caching, the network becomes a hybrid one which includes both traditional routers and cache-enabled routers. The hybrid routing departs significantly from the one in a complete cache enabled network. This makes the cache strategy more complex and the cooperation among neighboring routers more affected, which requires more fine-

J. Han, K. Xue, and J. Li are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China.

J. Zhang is with Science Island Branch of Graduate School, University of Science and Technology of China, Hefei, Anhui 230031, China.

Z. Huang is with the Institute of Space Integrated Ground Network, Hefei, Anhui 230088, China.

D. Wei is with the Computer and Information Science Department, Fordham University, Bronx, NY 10458, USA.

Corresponding Author: K. Xue (e-mail: kpxue@ustc.edu.cn).

grained algorithms.

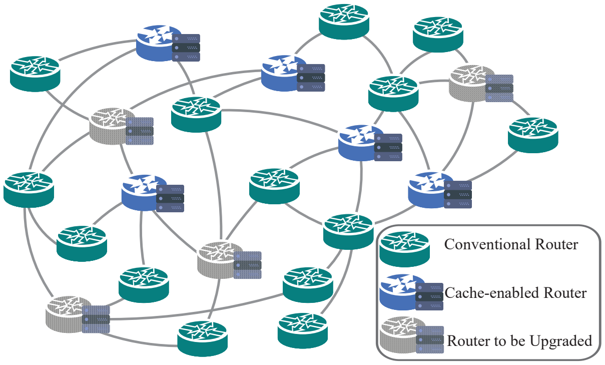


Fig. 1: The process of upgrading selective traditional routers to be cache-enabled ones.

For the first problem, we introduce a Selective Router Upgrade (SRU) problem, which aims to enhance the performance of a network by achieving low delay and high cache hit ratio. To address this objective, we propose two key optimizations, namely distance optimization and caching optimization of each router, which reflect to the improvement of content access delay and cache hit ratio in the network, respectively. In addition, most of previous works consider the scenarios for one-time deployment, but do not consider the hybrid network for progressively upgrading. SRU formulates a progressive upgrading process, the previously upgraded nodes will influence the future choice of nodes. SRU also considers the influence of upgraded nodes in a hybird network and presents a complete upgrading plan. Finally, due to the NP-hard nature of the SRU problem, we present an approximate solution using a modified greedy algorithm.

For the second problem, as the network evolves into a hybrid network with increasingly dense cache-enabled routers, simple non-cooperative caching schemes [15]–[17] could suffer from cache redundancy and result in poor caching performance. Although some cooperative caching strategies [18]–[20] are introduced. However, they work mostly on eliminating cache redundancy in return paths but fail to consider the impact of hybrid networks, where traditional routers may be present between cache-enabled routers. To address this issue, we introduce neighbor cooperative caching in the upgraded hybrid network. Based on the formulation of SRU problem, we then propose Local Replacement (LR) strategy and Neighbor Cooperative Caching (NCC) strategy, which considers both the cache hit ratio and content access delay in a local router and then modifies the cache strategy according to the situation of neighbor cooperation to improve the caching performance. By incorporating these strategies into the SRU ungraded network, we aim to achieve low delay and high hit ratio, thereby enhancing the overall performance of cache-enabled networks.

The main contributions of this paper can be summarized as follows:

- We put forward an SRU algorithm for upgrading selective traditional routers to be cache-enabled ones in hybrid cache-enabled networks. Both the influence of the routers upgraded in the past and different benefit metrics are considered to formulate the optimization problem. We prove that SRU problem is NP-hard and propose a $(1-1/e)$-approximation algorithm to solve it.
- For an upgraded network, we formulate the caching problem at the hybrid cache-enabled network. A simple non-cooperative LR strategy and a cooperative NCC strategy are proposed to effectively cache contents and reduce redundant caching in networks.
- The performance evaluation shows that SRU is able to select the nodes enabling the network to gain better performance to upgrade and NCC can further reduce users' access delay and improve cache hit ratio in the upgraded hybrid network.

The rest of this paper is organized as follows. Section II describes the background and Related Work. Sections III, IV, and V describe the SRU strategy, LR strategy, and NCC strategy, respectively. Section VI presents the evaluation results. Section VII concludes this paper.

II. Background and Related Work

A. Challenges of Enabling In-network Caching in Traditional IP Networks

Enabling in-network caching in traditional IP networks is a major direction of the future network, while such a hybrid network can both be compatible with traditional networks and provide better performance. In a hybrid network, just some of the routers in the network are enabled with the in-network caching function through the introduction of adding contents ID into a newly defined IP option. The existing routers do not need to be changed. Meanwhile, in the upgrading process, there are two main objectives to achieve. The first one is to select suitable routers to upgrade in a way that the network performance, in terms of content access delay and hit ratio, could be optimized. The second one is to find an effective caching strategy for cache-enabled routers in the upgraded hybrid network.

Due to budget constraints and the requirement for backward compatibility, only a subset of routers in the network will be selected to add caching functionality at each upgrade during the upgrading process. Therefore, the selection of the subset of routers is an important issue. With the upgrade of in-network caching enabled, a new problem arises. The number of cache-enabled routers is small, the distance between cache-enabled routers varies, and the cache space of cache-enabled routers in the network is limited. A caching strategy designed for a full cache-enabled network may not be suitable for a hybrid network. In this work, we model the contents cache behavior in the hybrid network, and further put forward a novel neighbor cooperative cache strategy to address this issue.

B. Related Works

1) Cache Location Strategies: Enable in-network caching in traditional IP networks is a major direction of the future network, while it can both be compatible with traditional networks and provide better performance. Detti *et al.* defined an IP option to make IP packets content-aware, named CONET [12]. We have proposed a new architecture called CAIP [13], which achieves in-network caching in current IP-based network architectures through the introduction of adding contents ID into a newly defined IP option. To enable in-network caching function in traditional IP networks, deciding where to deploy the cache-enabled routers has a critical impact on network performance. Some previous works [21]–[26] have worked on it. Although they are not designed for the previously mentioned architectures, the main idea is the same. Hasan *et al.* [21] studied the cache deployment optimization problem in content delivery networks (CDNs), which aims at minimizing the network cost while guaranteeing the delivery quality. Zhang *et al.* [23] proposed a cost-effective cache deployment problem for a two-tier HetNet in cellular networks. References [24]–[26] study the time-varying traffic demands to deploy cache space in large-scale WiFi systems. Most of the previous works consider the scenarios for one-time deployment, but do not consider the hybrid network for progressively upgrading. In this paper, we consider the overall benefits in the core network and present a complete upgrading plan for gradually upgrading the network.

2) Caching Strategies: Caching strategy determines the performance of a cache-enabled network [27]–[29]. Least Recently Used (LRU) [15] and Least Frequently Used (LFU) [17] policies are the most common content replacement cache strategies used in recent content routers, which evict the least recently used content and the least frequently used content, respectively, when the cache space overwhelms. LRU-sample [16] further improves its performance by sampling the caching operation. AdaptSize [30] considers the huge variance among content size and adjusts the cache parameters in the router adaptively, which can be more adapted to the network environment. Compared with the above-mentioned solutions that only work on a single router, cooperative caching is a more efficient and practical solution [31]–[33]. In recent research works, coordinated in-network caching strategies are introduced and can fall into two categories: i) On-path cooperative caching strategies [18], [20], [34], which control the content data along the return path. Lee *et al.* [34] provided T-caching, which utilizes the cooperation between content providers and routers to reduce the cache redundancy and improve cache hit performance. ii) Off-path cooperative caching strategies, such as Hash-Routing, SD-NCC [35]–[37], which control the content over neighboring routers within the neighborhood or AS. However, they do not consider the impact of the hybrid network. In our work, we consider the distances between content routers and the impact of neighboring routers to model the cache behavior, which is more suitable for a hybrid network.

III. Selective Router Upgrades

In this section, we elaborate SRU problem and the strategy of selecting a subset of routers to upgrade. SRU considers the influence of the upgraded routers and formulates a multi-objective optimization problem to achieve a higher cache hit ratio and lower access delay.

A. System Model

In our proposed system model, depicted in Fig. 2, the network topology is represented by a graph G, which comprises content providers, routers, and users. Content providers offer content that aligns with the interests of the users. The routers in the network are categorized into two distinct types: traditional routers and cache-enabled routers. Traditional routers within the network are responsible for packet forwarding functions, ensuring the efficient transfer of data packets. Cache-enabled routers possess the added capability of caching content and delivering it directly to the users. These cache-enabled routers enhance the performance of content delivery by reducing latency and network congestion, resulting in an improved user experience.

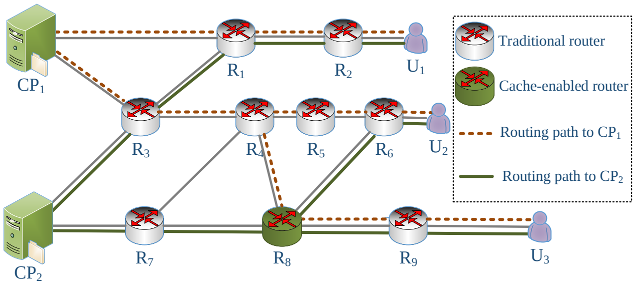


Fig. 2: An example of router upgrade.

To optimize the performance and efficiency of the system, we consider the process of router upgrades, where the decision on which routers to select for upgrading depends on the available budget. By carefully assessing the budget, we can determine the routers that will benefit the most from the upgrade, considering factors of network traffic and user demands.

We utilize the following notations to represent various elements within the network model. A router is denoted as R_j, where $j \in [1, n]$, and a content provider is denoted as CP_r, where $r \in [1, m]$. Users are grouped into different clusters, and each U_k has u_k users, where $k \in [1, q]$. The distance between router R_j and content provider CP_r is l_{jr}, where distance denotes the number-of-hop between two nodes.

To indicate whether a router R_j has been upgraded to a cache-enabled router, we introduce an upgraded indicator X_j, where $X_j \in \{0, 1\}$. If $X_j = 1$, it signifies that router R_j is a cache-enabled router, and vice versa. $\mathcal{D}$ denotes the set of routers that have already been upgraded, $X_j = 1$ if $j \in \mathcal{D}$. Once a router has been upgraded to a cache-enabled router, it should not undergo further upgrades in the future. P_j denotes the price of updating router R_j. The total cost of upgrading the routers should not exceed the budget which is denoted as B. We use p_r to denote the weight of content providers that relates to the proportion of contents provided by CP_r and $\sum_r p_r = 1$. The total amount and importance of content provided by different content providers vary, which may lead to different weights for different content providers.

To facilitate understanding, Table I summarizes the notations used in the system model.

TABLE I: The notations used in SRU

Notations	Meanings
R_j	router j
CP_r	content provider r
U_k	user cluster k
u_k	number of users in U_k
l_{jr}	distance between router R_j and content provider CP_r
X_j	upgraded indicator of router R_j
P_j	the price of updating router R_j
B	budget
$\mathcal{D}$	the set of routers that has already been upgraded
p_r	the weight of CP_r

B. Problem Formulation

After upgrading, the hybrid network is supposed to provide a lower delay and a higher cache hit ratio. However, there is a trade-off between the delay and the hit ratio. A router close to CP can serve more users and provide a higher hit ratio. On the other hand, a router located closer to users may provide a lower delay. As an example shown in Fig. 2, R_4 can serve U_2 and U_3 when the two users both require the contents from CP_1, and it has distance $l_{14} = 2$. R_5 can only serve U_2, but has distance $l_{13} = 3 > l_{14}$. We formulate an SRU problem that considers both of the two objectives and makes a trade-off between them. In the subsequent analysis, we define distance benefit as the improved performance gained from reducing delay, and define caching benefit as the improved performance gained from improving cache hit ratio.

Moreover, the benefit gained by an upgraded router is also affected by other upgraded routers, because the requests served by a downlink router will not be forwarded. As an example shown in Fig. 2, if R_9 is already upgraded, it will serve some requests from U_3. Therefore, the benefit gained from upgrading R_8 will be reduced. Due to the limited cache space of each cache-enabled router, the user's request has a probability ρ of being served, i.e., cache hit, and probability $(1-\rho)$ of forwarding a request to the next router. Let $u_{jr}^{sum}(\mathbf{X})$ denote the number of users that can be served at R_j when requesting contents from CP_r. We have:

$$u_{jr}^{sum}(\mathbf{X}) = \sum_{k \in \mathbb{U}_{jr}} u_k (1-\rho)^{\sum\limits_{j' \in \mathbb{R}_{jkr}} X_{j'}}, \tag{1}$$

where $\mathbf{X} = \{X_j \in \{0,1\}, \forall j \in [1,n]\}$ is the set of upgrade indicators. In our problem formation, we consider static routing paths between CP and users, where the routers forward interest packets and data packets along the shortest routing path between U_k and CP_r. Let CP_r-R_j-U_k denote a routing path, where R_j is a router that lies on the routing path between CP_r and U_k. We further use $\mathbb{R}_{jkr}$ to denote the set of routers that lie in the first subpath (CP_r-R_j) of the routing path CP_r-R_j-U_k, and $\mathbb{U}_{jr}$ to denote the set of user clusters that go through R_j to access CP_r.

The parameter ρ denotes the service probability on a cache-enabled router, which is related to the router's cache space and content popularity distribution. Considering that the content popularity is Zipf distribution with the shape parameter α. Therefore, the relative probability of a request for the i'th most popular content is proportional to A/i^{α}, where A is a constant related to α. There are N contents in total and each router can cache βN contents. Therefore, the upper bound of cache hit ratio is $\sum_{i=1}^{\beta N}(\frac{A}{i^{\alpha}})/\sum_{i=1}^{N}(\frac{A}{i^{\alpha}}) = \sum_{i=1}^{\beta N}(\frac{1}{i^{\alpha}})/\sum_{i=1}^{N}(\frac{1}{i^{\alpha}})$, which is related to that the router caches the most popular contents to full fill its cache space. Therefore, the user's request has an upper probability bound $\rho = \sum_{i=1}^{\beta N}(\frac{1}{i^{\alpha}})/\sum_{i=1}^{N}(\frac{1}{i^{\alpha}})$ of being served.

Let $a_j^r(\mathbf{X})$ and $b_j^r(\mathbf{X})$ denote the distance benefit and cache benefit of R_j for CP_r, respectively, we have:

$$\begin{aligned} a_j^r(\mathbf{X}) &= l_{jr} \cdot u_{jr}^{sum}(\mathbf{X}), \\ b_j^r(\mathbf{X}) &= u_{jr}^{sum}(\mathbf{X}), \end{aligned} \tag{2}$$

where $b_j^r(\mathbf{X})$ means that router R_j can provide cache space for $u_{jr}^{sum}(\mathbf{X})$ users, and $a_j^r(\mathbf{X}) = b_j^r(\mathbf{X}) \cdot l_{jr}$ represents the extent that R_j can serve $u_{jr}^{sum}(\mathbf{X})$ to keep contents away from the content provider for l_{jr}.

Therefore, for all the CPs, the total distance benefit and cache benefit of R_j are:

$$\begin{aligned} a_j(\mathbf{X}) &= \sum_{r=1}^{m} p_r \cdot a_j^r(\mathbf{X}), \\ b_j(\mathbf{X}) &= \sum_{r=1}^{m} p_r \cdot b_j^r(\mathbf{X}). \end{aligned} \tag{3}$$

To jointly optimize the delay and cache hit ratio, the SRU objective function is expressed as a multiple objectives function, which is defined as follows:

$$J_1(\mathbf{X}) = \gamma \sum_{j=1}^{n} X_j a_j(\mathbf{X}) + \theta \sum_{j=1}^{n} X_j b_j(\mathbf{X}), \tag{4}$$

where γ and θ are the weights to represent the importance of the terms.

A well-designed router upgrading solution should keep a balance between the cache hit and delay. Under such consideration, we make a simple normalization of the cache benefit and distance benefit by setting $\gamma = 1/\bar{l}$ and $\theta = 1$, where $\bar{l}$ is the mean value of all distance l_{jr}. Therefore, the cache benefit and distance benefit in the optimization problem have approximately the same mean value, which can provide balanced performance.

Let $V_j(\mathbf{X}) = \sum_r p_r\left(\gamma l_{jr} + \theta\right) u_{jr}^{sum}(\mathbf{X})$, then the objective function $J_1(\mathbf{X})$ can be expressed as:

$$J_1(\mathbf{X}) = \sum_{j=1}^{n} X_j V_j(\mathbf{X}). \tag{5}$$

And we model the SRU problem as follows.

SRU problem: Given a budget B, the objective of SRU is as follows:

$$\max_{\mathbf{X}} \quad J_1(\mathbf{X}) \tag{6}$$

$$s.t. \quad \sum_{j \in \mathcal{F}} X_j \cdot P_j \leq B, \tag{6a}$$

$$X_j = 1, \forall j \in \mathcal{D}, \tag{6b}$$

$$X_j \in \{0,1\}, \forall j \in \mathcal{F}, \tag{6c}$$

where $\mathcal{F} = \{j | j \in [1, n], j \notin \mathcal{D}\}$ denotes the set of routers which are not yet upgraded. Eq. (6a) indicates that the total upgrading price can not exceed the budget B. Eq. (6b) is the indicator of cache-enabled routers which have already been upgraded before this upgrading process. SRU only chooses the routers which are not in $\mathcal{D}$ to upgrade.

C. Selective Router Upgrades Strategy

It is hard to find the optimal solution for SRU problem in polynomial time. We give an approximate solution in this section.

Theorem 1. *The SRU problem is NP-Hard and the objective function of SRU problem is a non-decreasing submodular function.*

Proof. See Appendix. □

Since SRU problem is NP-hard, it is impossible to find the optimal result in polynomial time. However, as shown in [38], [39], a greedy heuristic algorithm can provides a $(1 - 1/e)$ approximate solution in polynomial time for maximizing submodular functions. Therefore, we provide Algorithm 1 to solve the SRU problem, which utilizes a modified greedy method to provide a $(1 - 1/e)$ approximation. In the mean loop of Algorithm 1, it calculates the new gained benefit of each non-upgraded router if it is upgraded as a cache-enabled router. Then it chooses a router with max value (which means the gained benefit divided by the price) to upgrade. The complexity of Algorithm 1 is $O(n^5)$, and n is the size of the network. Since the upgrade algorithm is an offline algorithm, which only needs to be run once before the upgrade, such a computational overhead is acceptable.

Algorithm 1: The Greedy Algorithm for SRU

```
Input:
  Network graph $G$,
  Budget: $B$,
  Price of routers: $\{P_j, j \in [1, n]\}$,
  The set of routers that have already been upgraded: $\mathcal{D}$.
Output:
  The router indicators: $\{\mathbf{X} | X_j \in \{0, 1\}, j \in [1, n]\}$.
1  /* $S_1$ denotes the set of all feasible solutions with one
     or two chosen routers */;
2  $\mathbf{X}_1 = \arg\max_{\mathbf{X} \in S_1} J_1(\mathbf{X})$;
3  /* $S_2$ denote the set of all feasible solutions with three
     chosen routers */;
4  $S_3 = \phi$;
5  for each $\mathbf{X}_0 \in S_2$ do
6      Initialize: $\mathbf{X} = \mathbf{X}_0$;
7      while $B \geq 0$ do
8          for each $j = 1$ to $n$ do
9              $V_j(\mathbf{X}) = \sum_r p_r \left(\gamma l_{jr} + \theta\right) u_{jr}^{sum}(\mathbf{X})$;
10             if $X_j = 1$ then
11                 $V_j^{add}(\mathbf{X}) = 0$;
12             else
13                 $V_j^{add}(\mathbf{X}) = V_j(\mathbf{X}) - V_j^s(\mathbf{X})$;
14         $j_{max} = \arg\max_j \frac{V_j^{add}(\mathbf{X})}{P_j}$;
15         $X_{j_{max}} = 1$;
16         $B = B - P_{j_{max}}$;
17     $S_3 = S_3 \cup \mathbf{X}$;
18 $\mathbf{X}_2 = \arg\max_{\mathbf{X} \in S_3} J_1(\mathbf{X})$;
19 $\mathbf{X}^* = \arg\max_{\mathbf{X}_1, \mathbf{X}_2} J_1(\mathbf{X})$;
20 return $\mathbf{X}^*$;
```

In line 13, SRU greedy algorithm chooses a router with max value of $V_j^{add}(\mathbf{X})/P_j$, where $V_j^{add}(\mathbf{X})$ is the gained benefit if R_j is upgraded as a new cache-enabled router:

$$V_j^{add}(\mathbf{X}) = J_1(\mathbf{X}^j) - J_1(\mathbf{X}), \tag{7}$$

where $\mathbf{X}^j$ denotes changing X_j to 1 in $\mathbf{X}$. The following shows the calculation of $V_j^{add}(\mathbf{X})$.

Let $\mathbb{R}_{jr}^s$ denote the set of routers along the route between R_j and CP_r. For a router $j' \in \mathbb{R}_{jr}^s$, if $R_{j'}$ is upgraded, it will lose a part of benefit $V_{j'r}^l(\mathbf{X})$:

$$V_{j'r}^l(\mathbf{X}) = \left(\gamma l_{j'r} + \theta\right) u_{jr}^{sum}(\mathbf{X}) \cdot \rho (1 - \rho)^{\sum_{j'' \in \mathbb{R}_{j'r}^s} X_{j''}}. \tag{8}$$

Therefore, we use $V_j^s(\mathbf{X})$ to denote the reduced benefit of all other cache-enabled routers if R_j is upgraded:

$$V_j^s(\mathbf{X}) = \sum_r p_r \sum_{j' \in \mathbb{R}_{jr}^s} X_{j'} V_{j'r}^l(\mathbf{X}). \tag{9}$$

Then, the true value of benefit that adds R_j as a new cache-enabled router will be:

$$V_j^{add}(\mathbf{X}) = V_j(\mathbf{X}) - V_j^s(\mathbf{X}). \tag{10}$$

The whole algorithm is divided into two phases. Let S_1 denote the set of all feasible solutions with one or two chosen routers, and S_2 denote the set of all feasible solutions with three chosen routers. In the first phase, the algorithm enumerates all feasible solutions in S_1 and finds $\mathbf{X}_1$ with the largest value of the objective function $J_1(\mathbf{X})$. In the second phase, the algorithm starts with each $\mathbf{X}$ in S_2 and completes each such set greedily and keeps the current solution feasible with respect to the knapsack constraint. Each time the algorithm recalculates $V_j^{add}(\mathbf{X})$ of the router that has not been upgraded and choose the router with the highest value until the budget is exhausted. Let $\mathbf{X}_2$ be the solution obtained in the second phase that has the largest value of the objective function over all choices of the starting set for the greedy algorithm. Finally, the algorithm outputs the best value between $\mathbf{X}_1$ and $\mathbf{X}_2$. It has been proven in [39] that Algorithm 1 provides an $(1 - 1/e)$-approximate solution at least.

The output $\mathbf{X}^*$ of Algorithm 1 denotes the set of all of the upgraded routers in the network, including the routers that have already been upgraded before this upgrading process. Therefore, $\mathbf{X}^* \setminus \mathcal{D}$ is the set of routers that are chosen to be upgraded in the current upgrading process.

IV. Local Replacement Strategy

In this section, we consider the caching behavior in a cache-enabled router. To simplify the description, we omit the subscript of router j and take the generalized behavior of a cache-enabled router. The cache size of the router is C. The size of a content i is c_i, and $x_i \in \{0,1\}$ is the indicator that denotes whether content i is cached in the router.

The same as the model of distance benefit and cache benefit of Section III, the distance benefit e_i and cache benefit f_i of caching a content i are:

$$\begin{aligned} e_i &= req_i \cdot L_i, \\ f_i &= req_i, \end{aligned} \tag{11}$$

where L_i denotes the saved distance, which is the distance from local router to the CP that provides content i, and the multiple objectives function is defined as follows:

$$J_2(\mathbf{x}) = \sum_i (\gamma e_i + \theta f_i) x_i, \tag{12}$$

where $\mathbf{x} = \{x_i\}$ is the vector of indicators.

Also, the optimization objective is to make a trade-off between minimizing the average delay and maximizing the cache hit ratio: $\max J_2(\mathbf{x})$ subject to $\sum\limits_i c_i x_i \leq C$. Let $v_i = (\gamma e_i + \theta f_i) = req_i \cdot (\gamma L_i + \theta)$ denote the total value of caching a content i, then $J_2(\mathbf{x}) = \sum_i v_i x_i$.

In the network, a router makes decision when receiving a new content. Let $\{i, i \in [1,n]\}$ denotes the contents that are already cached in the router. We formulate the local replacement problem as a $0-1$ Knapsack problem:

$$\max \quad \sum_{i=1}^{n} v_i x_i + v_{new} x_{new} \tag{13}$$

$$s.t. \quad \sum_{i=1}^{n} c_i x_i + c_{new} x_{new} \leq C, \tag{13a}$$

$$x_{new} \in \{0,1\}, \quad x_i \in \{0,1\}, \ \forall i, \tag{13b}$$

where $v_1, ..., v_n$, $c_1, ..., c_n$, and $x_1, ..., x_n$ are the cache value, content size, and indicator value of the contents cached in the router, v_{new}, c_{new}, and x_{new} are the cache value, content size, and indicator value of the new coming content. Constraint (13a) shows the contents cached in the local router should not exceed the overall cache size.

We use the number of requests for a content i in a past time interval to approximately estimate the req_i in the future. Assume that the distribution of users' interests is static over a period of time, then historically statistical data can provide a good reference to the trend of the future content popularity. To better estimate the content request number, local content req_i is defined as the number of requests to the content in time interval T. The cache-enabled router utilizes sliding window mechanism to dynamically calculate and update the req_i of contents. A cache-enabled router only records req_i of contents that have been requested within T. If content i has not been requested within T, the router sets $req_i = 0$ and deletes this record. If content i is first requested within T, the router sets $req_i = 1$. As shown in Fig. 3, when a router receives a new request for content i whose $req_i \geq 0$, req_i is updated to $req_i \leftarrow req_i \cdot (1 - \Delta t/T) + 1$, where Δt denotes the time interval between the last request of content i and the new coming request.

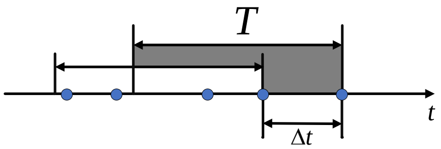


Fig. 3: Local content request.

When the router receives a new content and the cache space is full, it needs to decide whether to cache new content and which content in the cache memory should be replaced by the new content. To ensure fast processing speed within the router, LR strategy uses a classical greedy algorithm, as shown in Algorithm 2, to cope with the 0-1 Knapsack problem to find the approximate solution. All these input information can be easily calculated in a local router, which includes $\mathbf{c} = \{c_1, ..., c_n\}$ representing the set of the size of the cached contents and $\mathbf{v} = \{v_1, v_2, ..., v_n\}$ representing the set of the values of the cached contents. Through appropriate selection to meet the constraint condition and maximize the value of retained contents, we can obtain a set of retained contents after releasing enough space for the new entering content. The computational complexity of Algorithm 2 is $O(N)$ with a $(1/2)$ approximate solution, where N is the number of cached contents of a router. Especially, for a cache system whose contents have the same size, Algorithm 2 provides the optimal solution, and the time complexity of Algorithm 2 is $O(1)$. In most in-networking systems, content is divided into chunks with the same size, each of which is individually requested and delivered with a unique name. Therefore, cache-enabled routers typically route and cache with chunk-based granularity and the proposed Algorithm 2 provides an $O(1)$ solution with the optimal performance.

V. Neighbor Cooperative Caching Strategy

NCC strategy is put forward to make full use of neighbor nodes' cache resources. In an upgraded hybrid network, NCC should consider both the request number and the distance of neighboring cache-enabled routers to provide high hit ratio and reduced access delay via cooperation among neighboring routers.

A. Overview

In NCC strategy, the cache-enabled router keeps both the local cache index table and the neighbors' cache index table. To update the neighbor cache index table, a cache-enabled router periodically broadcasts its own cache index table and also receives the cache index tables broadcasted from its neighbors. When a cache-enabled router receives an interest packet, it first searches its local cache index table. If cache hit occurs, it sends the content to the user from its local cache. If not, the router then searches its neighbor cache index table, and forwards the interest packet to the neighbor if there is a cache hits at that neighbor. If the content is not found in the router's local cache index table nor in the neighbor cache index table, then the router will forward the interest packet to its upstream nodes. When a cache-enabled router receives a new content

Algorithm 2: Local Replacement Algorithm

```
Input:
Two sequence $\mathbf{v} = \{v_i, i = 1...n\}$ and
$\mathbf{c} = \{c_i, i = 1...n\}$,
The total cache capacity $C$,
The new content's size $c_{new}$ and cache value $v_{new}$.
Output:
The cache decision of the new content: $x \in \{0, 1\}$,
the evicted content set: $\mathbb{E}$.
for each $i = 1$ to $n$ do
    $p_i = v_i / c_i$;
$p_{new} = v_{new} / c_{new}$;
Sort by the ascending order $\mathbf{p} = \{p_{i_1}, p_{i_2}, ..., p_{i_n}\}$ and
  the corresponding size set for $P$ is
  $c = \{c_{i_1}, c_{i_2}, ..., c_{i_n}\}$;
if $p_{new} < p_{i_n}$ or $c_{new} > C$ then
    $x = 0$, $\mathbb{E} = \emptyset$;
    return $x$, $\mathbb{E}$;
else
    $c_{sum} = 0$, $p_{sum} = 0$;
    for each $j = 1$ to $n$ do
        $p_{sum} += p_{i_j}$;
        if $p_{sum} > p_{new}$ then
            $x = 0$, $\mathbb{E} = \emptyset$;
            return $x$, $\mathbb{E}$;
        $c_{sum} += c_{i_j}$;
        if $c_{sum} > c_{new}$ then
            $x = 1$, $\mathbb{E} = \{i_1, i_2, ...i_j\}$;
            return $x$, $\mathbb{E}$;
```

and replaces the old contents in its cache memory, it informs its neighbors about the evicted content. Then, a neighboring router can help cache the evicted data if it has sufficient cache memory.

There are two main considerations in NCC strategy. First of all, when responding to a user's request, a router treats its neighboring routers as an integrated one to search for the content. Therefore, the caching decision is affected by its neighbors' caching situation. Second, when a router is sending the evicted content to a neighbor, it also needs to find out which neighbor is the best choice. As shown in Fig. 4, the upgraded network is a hybrid one, which includes both the traditional routers and cache-enabled ones. The cache-enabled routers may not be physically connected directly, and there can be some traditional routers located in between two cache-enabled routers. NCC takes the different caching situations and the distance of neighbors into account to develop a modified LR strategy and a cooperative neighbor selection strategy to enhance the performance.

B. Modified Local Replacement Strategy

When a router gets a content and needs to decide whether it should be cached, the cache value is different from that of Section IV, because the router can get content not only from its cache, but also from its neighbors. Therefore, we remodel the cache value in NCC strategy while taking the neighbor popularity and distance into consideration.

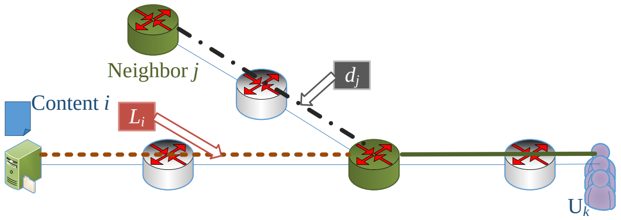


Fig. 4: In-network caching in a hybrid network.

As discussed in Section IV, the local cache value of content i is $v_i = req_i \cdot (\gamma L_i + \theta)$. However, a local router and its neighbor routers are considered to provide content for users at the same time in NCC strategy. In this situation, the neighbor requests should be considered in a local router. Meanwhile, neighbor distances in a hybrid network are different among different routers, which can also impact the cache value and need to be considered. Here we introduce a new concept d_j, which denotes the distance between the router and its neighbor j. Then, to modify the local cache value, we introduce two new concepts, repetitive cache value v_{ij}^r and collaborative cache value v_{ij}^c:

$$\begin{aligned} v_{ij}^r &= req_i \cdot (\gamma d_j + \theta), \\ v_{ij}^c &= req_i^j \cdot (\gamma (L_i^j - d_j) + \theta), \end{aligned} \tag{14}$$

where L_i^j is the saved distance of content i cached in router j, a cache-enabled neighbor, and req_i^j is the number of requests for content i cached in router j. L_i^j and req_i^j can be obtained when the router's neighbor requests content from it. If content i has already cached in router j, v_{ij}^r denotes the repetitive cache value of content i in the local cache-enabled router. It means that if a local router needs to obtain content i and its neighbor j has already cached content i, the local router can access content i from its neighbor, router j, and change saved distance from L_i to d_i. If a local router helps neighbor j cache content i, v_{ij}^c is the collaborative cache value of content i. The local router helps its neighbor j change the saved distance form L_i^j to d_j, that is to say the saved distance is reduced by $(L_i^j - d_j)$.

With NCC strategy, a router can get a content from its neighbors and also help its neighbors to get the content. Therefore, the cache value of content i is modified as:

$$w_i = \min\left\{v_i,\ \min_{j \in \mathbb{A}} v_{ij}^r\right\} + \sum_{j \in \mathbb{B}} v_{ij}^c, \tag{15}$$

where w_i is the revised cache value of content i in the local router. $\mathbb{A}$ is the set of neighbors that have cached content i, and $\mathbb{B}$ is the set of neighbors that need to get content i from the router. If a router's neighbor has cached the content, the content value decreases. Otherwise, if a router's neighbor needs to get the content from it, the content value increases.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3305195

The modified objective of local replacement problem is:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} w_i x_i + w_{new} x_{new} \\ s.t. \quad & (13a), (13b), \end{aligned} \tag{16}$$

where $w_1, ..., w_n$ and w_{new} are the modified cache value of cached contents and the new coming content.

To ensure linear processing speed within the router, modified LR strategy uses the same classical greedy algorithm as Algorithm 2 to cope with the 0-1 Knapsack problem by changing the set of the cached values $\mathbf{v} = \{v_1, v_2, ..., v_n\}$ to $\mathbf{w} = \{w_1, w_2, ..., w_n\}$.

C. Cooperative Caching and Neighbor Selection

When a local router has no available space to cache the content and has to evict some contents from its own cache, it can send the evicted content to its neighbor for cooperative caching. To reduce cache redundancy between neighbors and ensure maximum benefit, it only selects one neighbor to send the evicted content.

For simplicity of description, we illustrate the situation that content i_0 needs cooperative caching. For the situation with more than one content, we implement the same process for each piece of content in the descending order of cache values.

Let w_i^j denote the modified cache value of content i on neighbor j and $x_i^j \in \{0, 1\}$ denote the indicator of neighbor j. Then, the objective of cooperative cache replacement is formulated as:

$$\max \quad \sum_{j \in \mathbb{N}} \left(w_{i_0}^j x_{i_0}^j + \sum_{i \in \mathbb{I}_j} w_i^j x_i^j \right) \tag{17}$$

$$s.t. \quad c_{i_0} x_{i_0}^j + \sum_{i \in \mathbb{I}_j} c_i x_i^j \leq C_j, \ \forall j \in \mathbb{N}, \tag{17a}$$

$$\sum_{j \in \mathbb{N}} x_{i_0}^j \leq 1, \tag{17b}$$

where $\mathbb{N}$ is the set of neighboring routers, $\mathbb{I}_j$ is the set of contents cached in neighbor j, C_j is the cache space of neighbor j. Consider that a router needs linear-speed performance, we use a greedy algorithm to solve this problem. The neighbor selection algorithm is shown in Algorithm 3, where $j_0 = none$ means none of the neighbors is selected for cooperative caching. The computational complexity of Algorithm 3 is $O(NM)$ with a (1/2) approximate solution, where M is the number of neighbors. Especially, for a cache system whose contents have the same size, Algorithm 3 provides the best solution, and the time complexity of Algorithm 3 is $O(M)$.

When a router sends a content to the selected neighbor, the neighbor updates the content value and executes the cache replacement strategy. In addition, data caching in the neighbor may cause a chain reaction of data movement. If a large amount of data are unused or moved frequently, the performance will be degraded significantly. To avoid the chain reaction of data movement, for an evicted content, the neighbor does not perform the cooperative caching operation.

Algorithm 3: Neighbor selection

```
Input:
Information of content $i_0$: $L_{i_0}$, $req_{i_0}$, $c_{i_0}$,
The information of each neighbor $j \in \mathbb{N}$: $d_j$, $\{w_i^j\}$.
Output:
The selected neighbor content router: $j_0$.
for each $j \in \mathbb{N}$ do
    if $C_j - \sum_{i \in \mathbb{I}_j} c_i \geq c_{i_0}$ then
        return $j_0 = j$ ;
    $w_{i_0}^j += req_{i_0} \cdot (\gamma(L_{i_0} - d_j) + \theta)$;
    Run Algorithm 2 on neighbor content router $j$ and
      gets $x^j$, $\mathbb{E}^j$;
    $b_j = x^j w_{i_0}^j - \sum_{i \in \mathbb{E}^j} w_i^j$;
if $\max_j b_j \leq 0$ then
    return $j_0 = none$ ;
else
    return $j_0 = \arg\max_j b_j$ ;
```

VI. Performance Evaluation

In this section, we evaluate the performance of SRU, LR, and NCC. We use the BRITE topology generation tool to generate the test topology [40]. Based on Waxman's probability model [41], the topology consists of 1,000 routers with 0-10 end hosts per router and 1,000 object items. There are 2 CPs in the network and the contents are set to be evenly provided by the CPs, which means $p_r = 1/m = 1/2$ in the simulation. It should be noted that if the contents are not evenly provided by the CPs, the proposed algorithms can easily adapt to the new scenario by changing the setting of weight p_r. The data access pattern used in the simulation is Zipf distribution [42], and the arrival process of requests for objects at each access router follows a Poisson process.

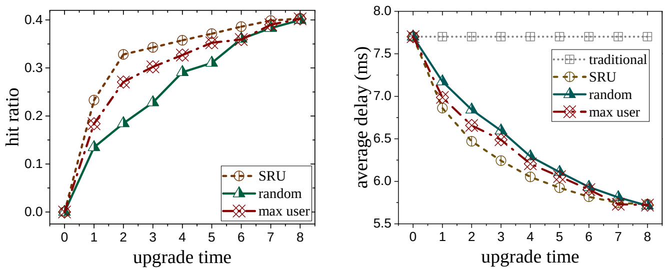
A. Performance of SRU

We compare the performance of SRU algorithm with random algorithm and max user algorithm, which randomly chooses routers and greedily chooses the router with the max access users, respectively. The cache-enabled router uses the basic caching algorithm Least Recently Used (LRU).

We first investigate the performance of different router upgrading schemes. The cache size of each cache-enabled router is set to 1% of the size of all contents. Zipf parameter α is setting to 1.0. We consider the case that a traditional network is progressively upgraded until all routers have been upgraded to cache-enabled ones. The process is divided into 8 upgrades, and each time it is allocated with the same budget, which is $1/8$ of the budget to upgrade all of the routers[1]. The average access delay and cache hit ratio, of different router upgrading mechanisms are respectively shown in Fig. 5.

[1] In reality, there is no need to repeatedly run our SRU algorithm until all of the routers in the network are upgraded to the cache-enabled ones. Depending on the budget, we could run SRU algorithm only once to upgrade only a subset of the traditional routers to gain the maximum benefit under the fixed budget.

This article has been accepted for publication in IEEE Transactions on Network Science and Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSE.2023.3305195

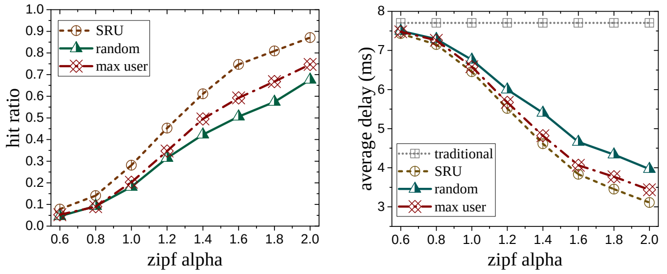


(a) Hit ratio.

(b) Average delay.

Fig. 5: Comparison of different router upgrading algorithms in different performance metrics at different upgrades.

The access delay of random upgrade almost goes down linearly, while the access delay of SRU and max user goes down faster at first and the rate of descent gradually slows down. The access delay of SRU and the max user is always smaller than the random method, and the gap between them is widening over time, which indicates that they provide better performance to users. The hit ratio of random upgrading also goes up linearly, while the hit ratios of SRU and max user go up faster at first, and then the rates of rising gradually slow down. Also, the hit ratio of SRU is always the highest among the three upgrading schemes.

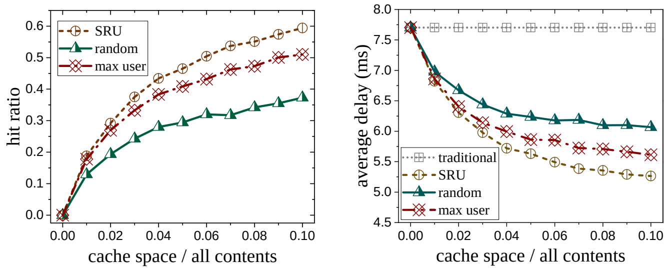


(a) Hit ratio.

(b) Average delay.

Fig. 6: Comparison of different router upgrading algorithms w.r.t. different Zipf parameters.

We then investigate the performance gained from the three router upgrading schemes with respect to the variation of Zipf parameter α changing from 0.6 to 2. The upgrading budget is set to $1/5$ of the price to upgrade all the routers and only upgrade the network for one time. The cache size of each cache-enabled router is set to 0.01 of the size of all contents. Fig. 6 shows the average access delay and cache hit ratio with different Zipf parameters. A higher Zipf parameter α represents a more concentrated distribution of content requests. It means that users tend to request some special contents while ignoring others. Therefore, a small cache space could provide a high hit ratio for the network. From Fig. 6(a) and Fig. 6(b), we can see that SRU provides higher performance in terms of average delay and cache hit ratio, respectively. With the increase of the Zipf parameter, the gap between SRU and the max user algorithm is also increasing. Because the max user algorithm ignores the impact of upgraded routers. If the upgraded routers are located on the same route, the requests are more likely to be served by a downstream router with a large Zipf parameter. SRU considers the impact of the upgraded routers, users, and distances, and thus provides higher performance.



(a) Hit ratio.

(b) Average delay.

Fig. 7: Comparison of different router upgrading algorithms w.r.t. different cache sizes.

We also investigate the performance gained by the three router upgrading schemes with respect to the variation of the router's cache size changing from 0.01 to 0.1 of the size of all the contents. Zipf parameter α is setting to 1.0. The upgrading budget is set to $1/5$ of the price to upgrade all the routers and only upgrade the network for one time. A larger cache space enables the router to cache more contents and thus provide a higher hit ratio. From Fig.7(a) and Fig. 7(b), we can see that SRU provides a higher performance in terms of average delay and cache hit ratio. With the increase of cache space, the gap between SRU and the max user algorithm is also increasing, which is because of the same reason as that in the case of a large Zipf parameter, i.e. the max user algorithm ignores the impact of upgraded routers.

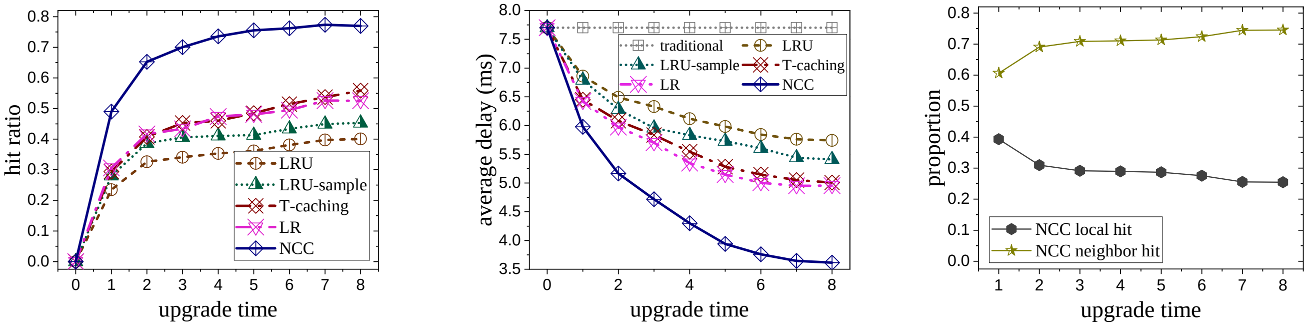
B. Performance of LR and NCC

In order to show the benefits of the LR and NCC strategies, we primarily focus on three key aspects:

- High hit ratio: By evicting the least popular content and increasing the availability of popular content in the cooperative neighbors, LR and NCC improve the hit ratio.
- Low delay: LR with local caching reduces the need for fetching content from distant servers. Cooperative caching, as employed by NCC, brings cached content closer to the end users, reducing network latency.
- Reduced cache redundancy: NCC facilitates content sharing and distribution among neighboring nodes, minimizing the duplication of cached content and promoting efficient cache utilization.

To evaluate the performance, we compare LR and NCC to the default LRU, LRU sample, T-caching schemes. In the simulation, we use “traditional” to denote a baseline of the traditional IP network, where there all the routers do not support the caching function.

We first investigate the performance of different caching schemes performed at different number of upgrades. The algorithms are repeated for 8 times, i.e. one time per router upgrading process, and for each time the used budget equals to $1/8$ of the price needed for upgrading all the routers. All

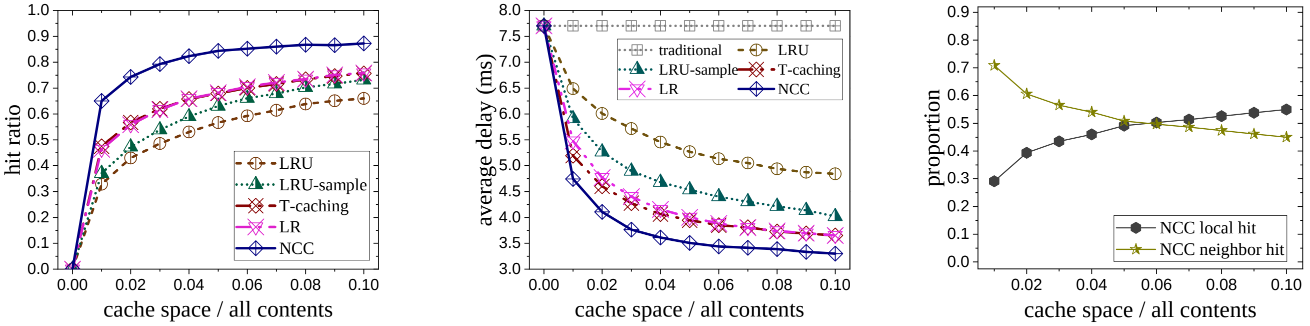


(a) Hit ratio.

(b) Average delay.

(c) Cache hit distribution.

Fig. 8: Comparison of caching schemes in different performance metrics at different number of upgrades.



(a) Hit ratio.

(b) Average delay.

(c) Cache hit distribution.

Fig. 9: Comparison of different caching schemes with different cache sizes.

the routers are upgraded at the end of the upgrade process. Zipf parameter α is setting to 1.0. The cache size of each cache-enabled router is set to 0.01 of the size of all contents. Fig. 8(a) and Fig. 8(b), respectively, show the average delay and cache hit ratio of LRU, LRU sample, T-caching, LR, and NCC at different times, where the sample ratio of LRU sample is 0.1. With the increase number of upgrades, the cache hit ratios of all the schemes are improved. Because there are more cache-enabled routers to provide in-network caching with more times of upgrading and the users can access content in the network more easily. Among different caching strategies, LRU provides the lowest hit ratio and the highest delay. LRU sample improves the hit ratio and delay by probabilistic caching and reducing redundancy. T-caching further improves the performance by the collaboration between content providers and routers. Compared with them, LR provides the similar performance to T-caching, and NCC provides more improvement based on LR and thus maintains the best performance. Because NCC has an increasing effect on eliminating redundancy among neighbors and requesting data from neighbors, it thus provides higher performance gain. Moreover, with more times of upgrading, the performance gain of NCC rendered to LR also increases. As the network continues to upgrade the routers, the content routers in the network become denser and denser and the distances between neighbor content routers become shorter and shorter. Therefore, getting content from one's neighbor will bring more benefit.

Fig. 8(c) shows the proportion of local hit and neighbor hit of NCC, where "NCC local hit" denotes the ratio of directly cache hits at cache-enabled routers, which means the cache hits are not requested by neighbor requests. "NCC neighbor hit" means the ratio of cache hits by the neighbor requests. We use $r_{local-hit}$ to denote "NCC local hit" and $r_{neighbor-hit}$ to denote "NCC neighbor hit", and they can be calculated as follows:

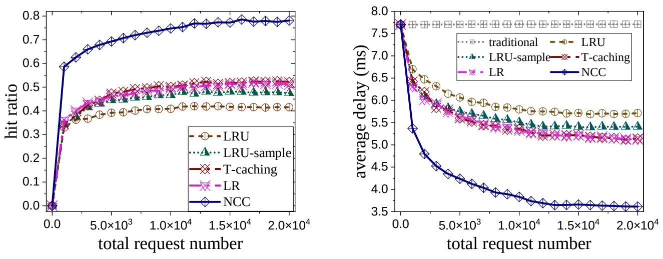
$$\begin{aligned} r_{local-hit} &= \frac{n_{local-hit}}{n_{all-hit}}, \\ r_{neighbor-hit} &= \frac{n_{neighbor-hit}}{n_{all-hit}}, \end{aligned} \tag{18}$$

where $n_{all-hit}$ denotes the number of all cache hits in the network, $n_{local-hit}$ denotes cache hits directly by original requests on local routers, and $n_{neighbor-hit}$ denotes cache hits by neighbor requests, $n_{local-hit} + n_{neighbor-hit} = n_{all-hit}$. Since NCC allows cache-enabled routers to get contents from their neighbors, it indirectly expands the capacity of routers. As shown in Fig. 8(c), more than 50% of cache hits are at neighbor routers. With the increase of cache-enabled routers in the network, the proportion of neighbor hit is also increasing, which means NCC is more likely to make cooperation in a highly upgraded network.

We then investigate the performance of different cache schemes with the variation of cache sizes. The upgrade budget is set to $1/5$ of the price needed for upgrading all routers and is only for one-time upgrading. The cache size of each router is changing from 0.01 to 0.1 of the size of all contents. Zipf parameter α is setting to 1.0. Fig. 9(a) and Fig. 9(b) show the average delay and cache hit ratio, respectively, of LRU, LRU sample, T-caching, LR, and NCC w.r.t. the increase of cache space, where the sample ratio of LRU sample is 0.1. The cache hit ratios of all the cache schemes show a growth trend, and NCC always has the highest cache hit ratio and provides the lowest delay.

Fig. 9(c) shows the proportion of local hit and neighbor hit of NCC. With the increase of cache size of cache-enabled routers, the proportion of neighbor hit is decreasing. Because a larger cache space means that a router can cache more contents and respond to users' requests by itself. If a router could cache all the contents in its cache space, it will not need cooperation from its neighbors. Even so, NCC still provides higher performance when the cache space is less than 0.1 of the size of all contents.

We also show the performance of different cache schemes with the variation of the number of requests. In this scenario, all the routers in the network are upgraded. The cache size of each cache-enabled router is set to 0.01 of the size of all contents. Zipf parameter α is setting to 1.0. Fig. 10 shows the average delay and cache hit ratio of LRU, LRU sample, T-caching, LR, and NCC, respectively. Among them, LRU only uses the information of the last request of contents, it soon gets the stationary value and only provides the lowest performance. On the contrary, others need more time to get the stationary value, because they consider more about the previous information of content requests and network topology, and can provide much better performance. Among them, NCC provides the highest hit ratio and the lowest delay. Moreover, NCC needs a little more time to get a higher steady reduction of access delay and a higher steady cache hit ratio. Because NCC needs more time to interact with neighbors and adjusts the cache space, which would indicate less cache redundancy among neighbors. The cache redundancy may reduce the hit ratio and degrade the performance. To eliminate unnecessary cache redundancy, NCC utilizes simple interactions between neighbors, and benefit more from the redundancy elimination.

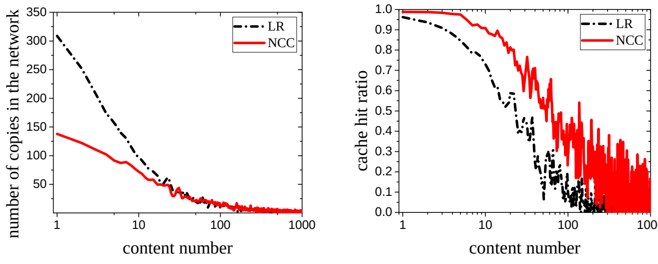


(a) Hit ratio.

(b) Average delay.

Fig. 10: Comparison of different caching schemes with different request numbers.

Fig. 11 shows the distribution of copies and hit ratio of contents in the network. The cache size of each cache-enabled router is set to 0.01 of the size of all contents. Zipf parameter α is setting to 1.0. Compared with LR, NCC reduces cache redundancy between routers and improves the diversity of contents that are cached in the network. For high popular contents ($i \leq 10$), NCC caches 50% fewer copies in the network than LR while still keeps a high cache hit ratio (close to 1). Therefore, NCC frees up more cache space to cache other contents and improves the overall cache hit ratio. Moreover, LR and NCC both provide a high hit ratio for high popular contents. With the help of neighbor collaboration, NCC provides a much higher hit ratio than LR when it comes to low popular content. In summary, NCC takes full advantage of the cache space through neighbor collaboration. It reduces cache redundancy between routers, and thus improves cache hit ratio, and reduces latency.



(a) Number of copies.

(b) Hit ratio.

Fig. 11: Distribution of copies and hit ratio of contents.

VII. Conclusion

In this paper, we presented a systematic solution for upgrading and caching in traditional IP networks, paving the way for improved content delivery and user experience in modern network architectures. In the process of router upgrading, depending on the budget and by an effective means, SRU carefully selects and upgrades a subset of routers while considering budget constraints and the impact on subsequent upgrades. We proved the NP-hardness of SRU and then gave a $(1 - 1/e)$-approximation upgrading algorithm. Based on SRU, LR and NCC are proposed for local caching and cooperative neighbor caching. LR optimizes cache hit ratio and content access delay locally, while NCC reduces cache redundancy and improves caching performance through neighbor cooperation. These strategies enhance overall network performance, offering higher hit ratios, lower latency, and reduced cache redundancy.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant No. 61972371, Youth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant No. Y202093, the Fundamental Research Funds for the Central Universities under Grant No. WK2100000039, and Xiaomi Young Talents Program.

References

[1] "Cisco annual internet report, (2018–2023) white paper," 2022, Cisco Visual Networking Index, Accessed on Aug. 2023. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2353–2371, 2018.

[3] C. Jiang, L. Gao, J. Luo, P. Zhou, and J. Li, "A game-theoretic analysis of joint mobile edge caching and peer content sharing," *IEEE Transactions on Network Science and Engineering*, vol. 22, no. 6, pp. 1445–1461, 2022.

[4] W. Yang, Y. Qin, Z. Yi, X. Wang, and Y. Liu, "Providing cache consistency guarantee for ICN-based iot based on push mechanism," *IEEE Communications Letters*, vol. 25, no. 12, pp. 3858–3862, 2021.

[5] Y. Han, R. Wang, and J. Wu, "Random caching optimization in large-scale cache-enabled internet of things networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 385–397, 2019.

[6] B. Nour, K. Sharif, F. Li, S. Biswas, H. Moungla, M. Guizani, and Y. Wang, "A survey of internet of things communication using ICN: A use case perspective," *Computer Communications*, vol. 142, pp. 95–123, 2019.

[7] L. Cui, E. Ni, Y. Zhou, Z. Wang, L. Zhang, J. Liu, and Y. Xu, "Towards real-time video caching at edge servers: A cost-aware deep Q-learning solution," *IEEE Transactions on Multimedia*, vol. 25, no. 1, pp. 302–314, 2021.

[8] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[9] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, "Network of information (NetInf)–an information-centric networking architecture," *Computer Communications*, vol. 36, no. 7, pp. 721–735, 2013.

[10] Q. N. Nguyen, R. Ullah, B.-S. Kim, R. Hassan, T. Sato, and T. Taleb, "A cross-layer green information-centric networking design toward the energy internet," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1577–1593, 2022.

[11] M. Conti, A. Gangwal, M. Hassan, C. Lal, and E. Losiouk, "The road ahead for networking: A survey on ICN-IP coexistence solutions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2104–2129, 2020.

[12] A. Detti, N. Blefari Melazzi, S. Salsano, and M. Pomposini, "CONET: A content centric inter-networking architecture," in *Proceedings of the 2011 ACM SIGCOMM workshop on Information-Centric Networking (ICN)*. ACM, 2011, pp. 50–55.

[13] K. Xue, T. Hu, X. Zhang, P. Hong, D. Wei, and F. Wu, "A withered tree comes to life again: Enabling in-network caching in the traditional IP network," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 186–193, 2017.

[14] Z. Luo, T. Li, Q. Zhang, R. Zhu, and T. Song, "A global name mapping system for ICN-IP coexistence," in *Proceedings of the 2022 ACM Conference on Information-Centric Networking (ICN)*, 2022, pp. 189–191.

[15] G. Quan, K. Ji, and J. Tan, "LRU caching with dependent competing requests," in *Proceedings of the 2018 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2018, pp. 459–467.

[16] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi, "Check before storing: What is the performance price of content integrity verification in LRU caching?" *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 59–67, 2013.

[17] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, 2002.

[18] B. Panigrahi, S. Shailendra, H. K. Rath, and A. Simha, "Universal caching model and markov-based cache analysis for information centric networks," *Photonic Network Communications*, vol. 30, no. 3, pp. 428–438, 2015.

[19] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "Wave: Popularity-based and collaborative in-network caching for content-oriented networks," in *Proceedings of the 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2012, pp. 316–321.

[20] Z. Ming, M. Xu, and D. Wang, "Age-based cooperative caching in information-centric networking," in *Proceedings of the 23rd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2014, pp. 1–8.

[21] S. Hasan, S. Gorinsky, C. Dovrolis, and R. K. Sitaraman, "Trade-offs in optimizing the cache deployments of CDNs," in *Proceedings of the 2014 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2014, pp. 460–468.

[22] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "Optimal cache allocation for content-centric networking," in *Proceedings of the 21st IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2013, pp. 1–10.

[23] S. Zhang, N. Zhang, P. Yang, and X. Shen, "Cost-effective cache deployment in mobile heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11 264–11 276, 2017.

[24] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, Y. Zhang, and X. Shen, "LEAD: Large-scale edge cache deployment based on spatio-temporal wifi traffic statistics," *IEEE Transactions on Mobile Computing*, vol. 20, no. 8, pp. 2607–2623, 2020.

[25] G. Tang, H. Wang, K. Wu, and S. Member, "Tapping the knowledge of dynamic traffic demands for optimal CDN design," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 98–111, 2019.

[26] F. Lyu, J. Ren, N. Cheng, P. Yang, M. Li, Y. Zhang, and X. Shen, "Demystifying traffic statistics for edge cache deployment in large-scale wifi system," in *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 965–975.

[27] C. Wang, C. Chen, Q. Pei, Z. Jiang, and S. Xu, "An information centric in-network caching scheme for 5G-enabled internet of connected vehicles," *IEEE Transactions on Mobile Computing*, vol. 22, no. 6, pp. 3137–3150, 2021.

[28] K. Kamran, A. Moharrer, S. Ioannidis, and E. Yeh, "Rate allocation and content placement in cache networks," in *Proceedings of the 2021 IEEE conference on computer communications (INFOCOM)*. IEEE, 2021, pp. 460–468.

[29] S. T. Thomdapu, P. Katiyar, and K. Rajawat, "Dynamic cache management in content delivery networks," *Computer Networks*, vol. 187, pp. 10–22, 2021.

[30] D. S. Berger, R. K. Sitaraman, and M. Harchol-Balter, "AdaptSize: Orchestrating the hot object memory cache in a content delivery network," in *Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 2017, pp. 483–498.

[31] Y. Guo, L. Duan, and R. Zhang, "Cooperative local caching under heterogeneous file preferences," *IEEE Transactions on Communications*, vol. 65, no. 1, pp. 444–457, 2016.

[32] H. Wu, J. Li, and J. Zhi, "Could end system caching and cooperation replace in-network caching in CCN?" in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (SIGCOMM)*, 2015, pp. 101–102.

[33] A. Gao, H. Liu, Y. Hu, W. Liang, and S. X. Ng, "Cooperative cache in cognitive radio networks: A heterogeneous multi-agent learning approach," *IEEE Communications Letters*, vol. 26, no. 5, pp. 1032–1036, 2022.

[34] S. Lee, I. Yeom, and D. Kim, "T-caching: enhancing feasibility of in-network caching in ICN," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 7, pp. 1486–1498, 2020.

[35] J. M. Wang, J. Zhang, and B. Bensaou, "Intra-AS cooperative caching for content-centric networks," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-Centric Networking (ICN)*. ACM, 2013, pp. 61–66.

[36] L. Saino, I. Psaras, and G. Pavlou, "Hash-routing schemes for information centric networking," in *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking (ICN)*. ACM, 2013, pp. 27–32.

[37] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software-defined networking, caching, and computing," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 274–287, 2018.

[38] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[39] M. Sviridenko, "A note on maximizing a submodular set function subject to a knapsack constraint," *Operations Research Letters*, vol. 32, no. 1, pp. 41–43, 2004.

[40] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proceedings of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MSWiM)*. IEEE, 2001, pp. 346–353.

[41] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, 1988.

[42] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proceedings of the 18th IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 1999, pp. 126–134.

Appendix

A. Proof of Theorem 1

Proof. For SRU problem, the objective can be written as $\max_{S\subseteq\mathcal{F}}\{f(S) : \sum_{j\in S} P_j \leq B\}$, where $f(S) = \sum_{j\in(S\cup\mathcal{D})} V_j(\mathbf{X})$. $f(S)$ can be divided into $f(S) = \sum_r p_r(\gamma f_1^r(S) + \theta f_2^r(S))$, where $p_r > 0, \gamma > 0, \theta > 0$, and $f_1^r(S) = \sum_{j\in(S\cup\mathcal{D})} l_{jr}u_{jr}^{sum}(\mathbf{X})$, $f_2^r(S) = \sum_{j\in(S\cup\mathcal{D})} u_{jr}^{sum}(\mathbf{X})$.

Let $m_{jkr}^{sum}(\mathbf{X}) = u_k(1-\rho)^{\sum_{j'\in\mathbb{R}_{jkr}} X_{j'}}$, we have $u_{jr}^{sum}(\mathbf{X}) = \sum_{k\in\mathbb{U}_{jr}} m_{jkr}^{sum}(\mathbf{X})$.

When $y \in \mathcal{F} \setminus S$, let $\mathbf{X}^y = \{X_j^y\}$ denote the upgrade indicators of $(S \cup y)$. Let $\Theta = \mathbb{R}_{yr}^s \cap (S \cup \mathcal{D})$ denote the set of upgraded routers along the route between R_y and CP_r. we first show the proof of the submodular property of $f_1^r(S)$. The $j \in (S \cup \mathcal{D} \cup y)$ can be divided into three cases:

1) For $j \in (S \cup \mathcal{D}) \setminus \Theta$, adding the new cache-enabled router y does not change the value of R_j. Then, we have $u_{jr}^{sum}(\mathbf{X}^y) = u_{jr}^{sum}(\mathbf{X})$ and $V_j(\mathbf{X}^y) = V_j(\mathbf{X})$.

2) For $j \in \Theta$, adding the new cache-enabled router y makes the value of R_j decrease. We have

$$\begin{aligned} m_{jkr}^{sum}(\mathbf{X}^y) &= u_k(1-\rho)^{\sum_{j'\in\mathbb{R}_{jkr}} X_{j'}^y} \\ &= \begin{cases} (1-\rho)m_{jkr}^{sum}(\mathbf{X}), & k \in \mathbb{U}_{yr}, \\ m_{jkr}^{sum}(\mathbf{X}), & k \notin \mathbb{U}_{yr}. \end{cases} \end{aligned}$$

And

$$\begin{aligned} & u_{jr}^{sum}(\mathbf{X}) - u_{jr}^{sum}(\mathbf{X}^y) \\ &= \sum_{k\in\mathbb{U}_{jr}} \left(m_{jkr}^{sum}(\mathbf{X}) - m_{jkr}^{sum}(\mathbf{X}^y)\right) \\ &= u_{yr}^{sum}(\mathbf{X})\rho(1-\rho)^{\sum_{j'\in\mathbb{R}_{jkr}\setminus\mathbb{R}_{ykr}} X_{j'}^y}. \end{aligned}$$

3) For y, we have $X_{j'}^y = X_{j'}$, $\forall j' \neq y$, therefore $a_y^r(\mathbf{X}^y) = a_y^r(\mathbf{X}) = l_{yr}u_{yr}^{sum}(\mathbf{X})$.

Then, we can calculate the increase value of adding y as a newly chosen router above S:

$$\begin{aligned} & f_1^r(S \cup y) - f_1^r(S) \\ =& a_y^r(\mathbf{X}^y) - \sum_{j\in\Theta}\left(a_j^r(\mathbf{X}) - a_j^r(\mathbf{X}^y)\right) \\ =& u_{yr}^{sum}(\mathbf{X})\left(l_{yr} - \sum_{j\in\Theta} l_{jr}\cdot\rho(1-\rho)^{\sum\limits_{j'\in\mathbb{R}_{jkr}\setminus\mathbb{R}_{ykr}} X_{j'}}\right), \end{aligned}$$

where $0 \leq \rho \leq 1$, and $l_{jr} < l_{yr}$ if $j \in \Theta$. Therefore, $f(S \cup y) - f(S) \geq u_{yr}^{sum}(\mathbf{X})l_{yr}\left(1 - \rho\sum_{i=0}^{|\Theta|-1}(1-\rho)^i\right)$. Then we have $f(S \cup y) - f(S) \geq 0$.

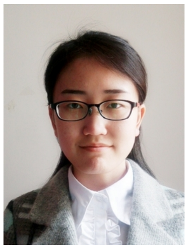
Let $z \in \mathcal{F} \setminus (S \cup y)$. When adding y to $S \cup z$, we have

$$\begin{aligned} & f_1^r(S \cup z \cup y) - f_1^r(S \cup z) \\ =& u_{yr}^{sum}(\mathbf{X}^z)\left(l_{yr} - \sum_{j\in\Theta} l_{jr}\cdot\rho(1-\rho)^{\sum\limits_{j'\in\mathbb{R}_{jkr}\setminus\mathbb{R}_{ykr}} X_{j'}^z}\right), \end{aligned}$$

where $u_{yr}^{sum}(\mathbf{X}^z) \leq u_{yr}^{sum}(\mathbf{X})$, and $\sum_{j'\in\mathbb{R}_{jkr}\setminus\mathbb{R}_{ykr}} X_{j'}^z \geq \sum_{j'\in\mathbb{R}_{jkr}\setminus\mathbb{R}_{ykr}} X_{j'}$. Therefore, we have $f_1^r(S \cup y) - f_1^r(S) \geq f_1^r(S \cup z \cup y) - f_1^r(S \cup z)$. Then $f_1^r(S)$ is a nondecreasing submodular function.

And once again, we can prove that $f_2^r(S)$ is a nondecreasing submodular function. Since $f(S) = \sum_r p_r(\gamma f_1^r(S) + \theta f_2^r(S))$, $f(S)$ is a nondecreasing submodular function.

As proof in [38], maximizing the submodular set functions are NP-hard problems. Therefore, SRU problem is NP-hard. □



Jiangping Han (M'22) received her bachelor's degree and the doctor's degree both from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2016 and 2021, respectively. From Nov. 2019 to Oct. 2021, She was a visiting scholar with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University. She is currently a Post-Doctoral researcher with the School of Cyber Science and Technology, USTC. Her research interests include future Internet architecture design and transmission optimization.

Kaiping Xue (M'09-SM'15) received his bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his doctor's degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a postdoctoral researcher with the Department of Electrical and Computer Engineering, University of Florida. Currently, he is a Professor in the School of Cyber Science and Technology, USTC. He is also a director of Network and Information Center, USTC. His research interests include next-generation Internet architecture design, transmission optimization and network security. His work won best paper awards in IEEE MSN 2017 and IEEE HotICN 2019, the Best Paper Honorable Mention in ACM CCS 2022, the Best Paper Runner-Up Award in IEEE MASS 2018, and the best track paper in MSN 2020. He serves on the Editorial Board of several journals, including the IEEE Transactions on Dependable and Secure Computing (TDSC), the IEEE Transactions on Wireless Communications (TWC), and the IEEE Transactions on Network and Service Management (TNSM). He has also served as a (Lead) Guest Editor for many reputed journals/magazines, including IEEE Journal on Selected Areas in Communications (JSAC), IEEE Communications Magazine, and IEEE Network. He is an IET Fellow and an IEEE Senior Member.

Jian Li (M'20) received his bachelor's degree from the Department of Electronics and Information Engineering, Anhui University, in 2015, and received doctor's degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2020. From Nov. 2019 to Nov. 2020, he was a visiting scholar with the Department of Electronic and Computer Engineering, University of Florida. From Dec. 2020 to Dec. 2022, he was a Post-Doctoral researcher with the School of Cyber Science and Technology, USTC. He is currently a research associate with the School of Cyber Science and Technology, USTC. He also serves as an Editor of China Communications. His research interests include wireless networks, next-generation Internet, and quantum networks.

Jing Zhang received the B.S. degree in Electronic Information Science and Technology from Hefei University of Technology, China, in 2007 and the M.S. degree in Control Theory and Control Engineering from Anhui University of Science and Technology, Huainan, in 2010. From 2017 to 2021, he was a communication algorithm engineer with the 38th Research Institute of China Electronics Technology Group Corporation, Hefei. He is currently a Ph.D. student with Science Island Branch, Graduate School of USTC, Hefei. His research interests include space information networks, satellite communication system design, and Ad-hoc networks.

Zixuan Huang received the bacholar’s degree in Electronic Information Science and Technology from Hefei University of Technology, Hefei, China, in 2016 and the M.S. degree in Electrical and Information Engineering from Harbin Institute of Technology, Shenzhen, in 2019. From 2019 to 2021, she was a communication algorithm engineer with Huawei, ShenZhen and Shanghai. She is currently a communication algorithm engineer with Institute of Space Integrated Ground Network, Hefei, China. Her research interests include space information networks, err control codes, and Ad-hoc networks.

David S.L. Wei (SM’07) received his Ph.D. degree in Computer and Information Science from the University of Pennsylvania in 1991. From May 1993 to August 1997 he was on the Faculty of Computer Science and Engineering at the University of Aizu, Japan (as an Associate Professor and then a Professor). He has authored and co-authored more than 140 technical papers in various archival journals and conference proceedings. He is currently a Professor with the Computer and Information Science Department at Fordham University. He was a lead guest editor or a guest editor for several special issues in the IEEE Journal on Selected Areas in Communications, the IEEE Transactions on Cloud Computing, and the IEEE Transactions on Big Data. He also served as an Associate Editor of IEEE Transactions on Cloud Computing, 2014-2018, an editor of IEEE J-SAC for the Series on Network Softwarization & Enablers, 2018 – 2020, and an Associate Editor of Journal of Circuits, Systems and Computers, 2013-2018. Due to his research achievements in information security, Dr. Wei is the recipient of IEEE Region 1 Technological Innovation Award (Academic), 2020, for contributions to information security in wireless and satellite communications and cyber-physical systems. He is a member of ACM and AAAS, and is a life senior member of IEEE, IEEE Computer Society, and IEEE Communications Society. Currently, Dr. Wei focuses his research efforts on cloud and edge computing, cybersecurity, and quantum computing and communications.