network

MDPI

Review

Wireless Caching: Making Radio Access Networks More than Bit-Pipelines

Wei Chen [1,*] and H. Vincent Poor [2]

[1] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2] Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA; poor@princeton.edu
[*] Correspondence: wchen@tsinghua.edu.cn

Abstract: Caching has attracted much attention recently because it holds the promise of scaling the service capability of radio access networks (RANs). We envision that caching will ultimately make next-generation RANs more than bit-pipelines and emerge as a multi-disciplinary area via the union with communications, pricing, recommendation, compression, and computation units. By summarizing cutting-edge caching policies, we trace a common root of their gains to the prolonged transmission time, which is then traded for higher spectral or energy efficiency. To realize caching, the physical layer and higher layers have to function together, with the aid of prediction and memory units, which substantially broadens the concept of cross-layer design to a multi-unit collaboration methodology. We revisit caching from a generalized cross-layer perspective, with a focus on its emerging opportunities, challenges, and theoretical performance limits. To motivate the application and evolution of caching, we conceive a hierarchical pricing infrastructure that provides incentives to network operators and users. To make RANs even more proactive, we design caching and recommendation jointly, showing a user what it might be interested in and what has been done for it. Furthermore, the user-specific demand prediction motivates edge compression and proactive MEC as new applications. The beyond-bit-pipeline RAN is a paradigm shift that brings with it many cross-disciplinary research opportunities.

Keywords: caching; radio access networks; virtual real-time services; AI-empowered networks; cross-layer design; pricing; recommendation; edge compression; proactive computation

check for updates

Citation: Chen, W.; Poor, H.V. Wireless Caching: Making Radio Access Networks More than Bit-Pipelines. *Network* **2021**, *1*, 146–164. https://doi.org/10.3390/network1020010

Academic Editor: Amitava Datta

Received: 27 June 2021
Accepted: 2 August 2021
Published: 16 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Modern radio access networks are capable of achieving data rates of Gbps, while they may still fail to meet the predicted bandwidth requirements of future networks. A recent report from Cisco [1] forecasts that mobile data traffic will grow to 77.49 EB per month in 2022. In theory, a human brain may process up to 100T bits per second [2]. As a result, a huge gap may exist between the future bandwidth demand and provision in next generation radio access networks (RANs). Unfortunately, on-demand transmission that dominates current RAN architectures has almost achieved its performance limits revealed by Shannon in 1948, given extensive development of physical layer techniques in the past decades. On the other hand, the radio spectrum has been over-allocated, while the overall energy consumption is explosive. Since the potential of on-demand transmission has been fully exploited, it is time to conceive novel transmission architectures for sixth generation (6G) networks [3] so as to scale its service capability. The cache-empowered RAN is one of the potential solutions that hold the promise of scaling service capability [4].

Caching techniques were originally developed for computer systems in the 1960s. Web caching was conceived for the Internet due to the explosively increasing number of websites in the 2000s. In contrast to on-demand transmission, caching allows proactive content placement before being requested, which has motivated some novel infrastructures such as information-centric networks (ICNs) and content delivery networks (CDNs).

More recently, caching has been found to substantially benefit data transmissions over harsh wireless channels and meet growing demands with restrained radio resources in various ways [5–8]. For instance, caching is capable of not only exploiting the idle spectrum to offload and reduce the peak-time bandwidth requirement, but also enabling physical-layer multicasting in the content placement phase. With coded caching [9], the physical-layer multicasting gain can be attained in the delivery phase even when users request different files. Caching also enables novel interference cancellation and alignment schemes that hold the potential of substantially improving the device density in RANs [10]. In short, caching exploits scalable storage resources to improve the highly restrained radio resource efficiency. One cost of caching involves the memories at edge nodes and end devices, whose capacities have grown in the past few decades due to technological advances. The various caching gains are attracting increasing attention from both academia and industry in the era of 6G research. For instance, a focus group on machine learning for future networks including 5G (ML5G) was launched by ITU in 2019 with a key purpose of enabling caching.

Though considerable literature on the subject of wireless caching exists, there is a need to revisit it from a cross-layer perspective, as shown in Figure 1. Why? A simple answer is that caching relies on both the content placement that transmits on the minus time axis before user requests, and content delivery that transmits upon user requests. Therefore, multiple layers of a RAN have to function together to realize caching. Furthermore, caching relies on the coordination of wireless communication, prediction, and the use of memory units, which should be jointly scheduled in the time domain. It is insufficient to make binary decisions on “to cache or not to cache” in practice. By contrast, a RAN should determine when and how long a content item is cached, as well as how it is transmitted in the placement and delivery phases, etc. In summary, a systematic discussion of caching with multi-layer or even multi-unit collaborations is necessary. More importantly, few studies focus on changing the underlying design methodologies of RANs to fully exploit caching. In particular, caching will make next-generation RANs much more proactive and become more than bit-pipelines. It motivates a proactive infrastructure that integrates wireless transceivers, request predictors, and cache memories. This infrastructure predicts what and when a user requests or schedules, when and how to push, and determines how long to cache by itself. As a result, there is a need to revisit caching, and in this paper we do so from the following three perspectives.

- From a time-domain perspective, a unified view of caching gains and costs is presented. Pushing cacheable information objects prolongs their transmission times significantly while serving users within their tolerances. To adopt various wireless techniques that trade the transmission time for spectral and/or energy efficiency gains, user-specific request prediction is particularly beneficial, based on which a beyond-bit-pipeline RAN proactively schedules what, when, and how to push based on its own prediction of users’ requests. Some fundamental limits of cache-empowered RANs can be revealed from a cross-layer perspective.
- From an incentive perspective, a pricing mechanism is conceived to advocate the application and evolution of caching policies. Since no one is willing to pay for what he or she never reads, cache-empowered RANs abandon the bit-pipeline-oriented pricing that charges a user according to her or his throughput or access time. We envision a hierarchical framework to price caching services, which creates incentives for RANs to foresee future requests, push the right files, and participate required collaborations.
- From a transparency perspective, recommendation systems (RSs) make cached files visible to users. A cache-friendly RS tells a user not only what he or she might be interested in, but also what has been done for her or him. If the user accepts the recommendation to read the cached file, he or she enjoys low latency and possibly a reduced service fee because caching saves radio resources. A RAN with joint caching and recommendation will no longer serve as a bit-pipeline only, since it manipulates

users. Furthermore, preference-aware data compression, also referred to as edge compression, is made practical by learning and understanding individual users. The idea of proactive caching can be also borrowed to offload peak-time tasks of mobile edge computing (MEC). The merging of communication, computation, compression, and recommendation makes RANs more than bit-pipelines.

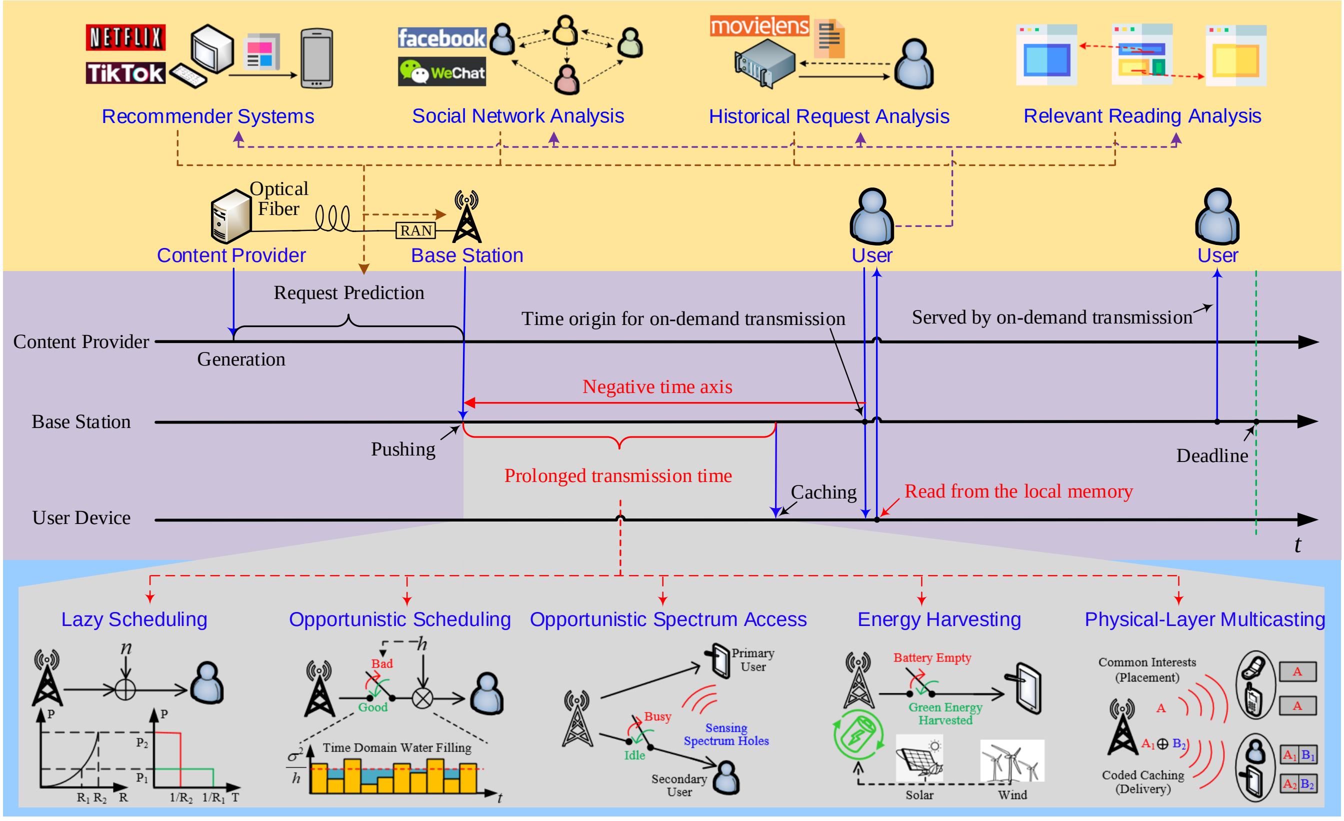


Figure 1. A unified framework for understanding caching gains from a time-domain perspective. Caching prolongs the transmission time, which enables various wireless techniques e.g., in Table 1 that trade the transmission time for energy and/or spectral efficiencies.

In this article, we begin our discussion with gains, costs, and needs of cache-empowered RANs that determine what, when, and how to transmit and cache without waiting for requests from users. Then, a more precise prediction that foresees future user request in the time domain is reviewed. Based on the request time prediction, joint scheduling of communications links and cache memories is enabled to optimize when and how content items are cached. In particular, we review the energy-, spectral-, and memory-efficient caching methods with focus on their theoretical performance limits. To advocate caching and user cooperation therein, a novel hierarchical pricing architecture is also discussed. Finally, we envision that caching will merge several application layer roles, such as recommender systems, data compression, and MEC, thereby holding the potential of making RANs more than bit-pipelines.

Table 1. Tradeoff Between the Transmission Time and SE/EE.

Transmission Techniques	Application Scenarios	How Is SE or EE Gain Attained?	Why Is Delay Increased?
Lazy Scheduling	Additive White Gaussian Noise Channels	Due to the convexity of Shannon capacity, EE is a decreasing function of the transmission power/rate.	Low data rate
Opportunistic Scheduling	Fading Channels	EE/SE is increased by time domain water-filling, or simply accessing good channels only.	Channel states remaining poor
Opportunistic Spectrum Access	Secondary Users	SE is increased by sensing and accessing idle timeslots or spectrum holes.	Spectrum remaining busy
Energy Harvesting	Renewable Energy Powered BSs/UEs	The renewable energy harvested from solar panels, wind turbines, or even the RF environment helps to save grid power.	No or little energy harvested
Physical-Layer Multicasting	Users with Common Interests	Multiple users located in the same cell are served by broadcasting a common signal to them.	Waiting for common requests

2. Proactive Service: Gains, Costs, and Needs

Without waiting for users' orders, a cache-empowered RAN provides proactive services. We first review how caching benefits RANs , what it has to pay, and what is required. From both communication and memory perspectives, we come to a conclusion that a binary decision on "to cache or not to cache" is insufficient [11].

2.1. Caching Gains: A Time-Domain Perspective

Revisiting wireless caching gains is critical in the optimization of cache-empowered RANs. Those gains reported in the existing literature can be mainly cast into three main categories.

- Caching enables physical layer multicasting [12]. In theory, caching is capable of serving infinitely many users with a common request, thereby making RANs scalable. Classic on-demand transmission can seldom benefit from the multicasting gain because users seldom ask for a common message simultaneously. Aligning common requests in the time domain may, however, cause severe delay and damage Quality-of-Service (QoS). Proactive caching brings a solution to attain multicasting gain without inducing delay in data services. Even when users have different requests, judiciously designed coded caching strategies [13,14] allow RANs to enjoy the multicasting gain.
- Caching extends the tolerable transmission time, thereby bringing spectral efficiency (SE) or energy efficiency (EE) gains. Lazy scheduling [15], opportunistic scheduling [16,17], opportunistic spectrum access (OSA) [18], and energy harvesting (EH) [19] may increase the SE and EE. However, their applications are usually prohibited or limited due to their random transmission delay. Caching enables content transmission before user requests and hence substantially prolongs the delay tolerance.
- Caching enables low-complexity interference mitigation or alignment [10]. It is well known that a user can cancel a signal's interference based on prior knowledge about the message that the signal bears. Classic successive interference cancellation (SIC) decodes the interference first by treating the desired signal as noise. However, SIC can suffer from high complexity and error propagation. By contrast, caching provides reliable prior knowledge on the interfering signal, which significantly reduces the complexity of interference cancellation.

We present a unified time-domain framework to trace the common root of caching gains revealed by a large body of research. In many wireless techniques, there exists a fundamental tradeoff between transmission time and radio resource efficiency, as summarized in Table 1. In practice, however, trading the transmission time for energy efficiency (EE) or spectral efficiency (SE) is mostly prohibited due to the reactive service mode. When content items are transmitted upon user requests, an increased transmission time results in a large delay, leading to poor QoS.

Caching is a straightforward solution prolonging the transmission time by allowing RANs to push files to the network edge or a user's device before they are requested. In the reactive mode, a user's request time is usually regarded as the time origin of scheduling. Hence content placement is launched on the negative time-axis. In this way, we increase the transmission time while assuring that a user experiences small delay, as shown in Figure 1. In this case, the methods in Table 1 can be exploited to increase the EE or SE. Though their data rate is low or unstable, caching allows a user to experience real-time services virtually.

Caching is expected to benefit the next generation RAN, e.g., 6G, in many aspects. A natural question to ask is which layer caching belongs to. If it only makes binary decisions on "to cache or not to cache" in each node, it is more like a network-layer protocol. In other words, a binary decision on "to cache or not to cache" is insufficient. By contrast, a more precise decision on "when and how to cache" is desired, which requires caching to function with the physical and link layers. In particular, since the data rate of any wireless link is bounded, neither content placement nor delivery can be accomplished immediately. The content placement and delivery tasks must be carefully scheduled in the time domain. For content placement or delivery over time-varying channels, radio resource allocation should adapt to instantaneous channel states, while meeting the deadlines. Furthermore, as the cache-enabled multicasting or interference cancellation requires concurrent transmission, asynchronous transmissions should be aligned in the time domain. In summary, separate caching from the physical and link layers is not possible.

Instead of waiting for a user's command, a RAN itself not only makes binary decisions on whether "to cache or not to cache", but also determines "when and how to cache". Thus, it requires novel functions beyond a bit-pipeline. In [15–17], threshold-based policies avoid pushing undemanded files that waste energy. When the demand probability for a content is below a certain threshold, its caching gain fails in compensating for the waste of radio resources. In addition, for popular files, caching too early results in the reduction of the available time for content pushing, resulting in the loss of SE/EE gains. Caching too late, however, may miss the request. To make caching practical, careful scheduling is desired in both placement and delivery phases. The time-domain framework in Figure 1 implies unified resource allocation that optimizes caching gains subject to the latency requirement. The framework can be applied with an arbitrary physical-layer scheme given its instantaneous rate-cost function. Moreover, by tracing the common root of caching gains, we may foresee how emerging wireless techniques with potentially large delay can be enabled by caching.

2.2. A Case Study: Caching-Enhanced mmWave Coverage

Caching enables a low-cost and flexible deployment of base stations (BSs) working in mmWave or even Thz bands. Transmissions over extremely high frequency bands provide much wider bandwidth and much higher throughput but suffer from poor coverage due to severe path loss and blocking. To overcome this, massive multiple-input multiple-output (MIMO) and intelligent reflecting surfaces (IRS) have been extensively studied.

Caching provides an alternative solution [20]. With caching, mmWave BSs do not necessarily cover every corner of the whole area. As shown in Figure 2, mobile users enter and leave the area covered by mmWave signals dynamically. When high-speed access is available, a user will cache, as much as possible, popular content in which it may be interested. After leaving the mmWave covered area, the user may find its requests partially served by its local cache. The cache-empowered hierarchical RAN improves the user's QoS without dense deployment of mmWave BSs. Similarly, this idea is helpful when there is a human body blocking effect.

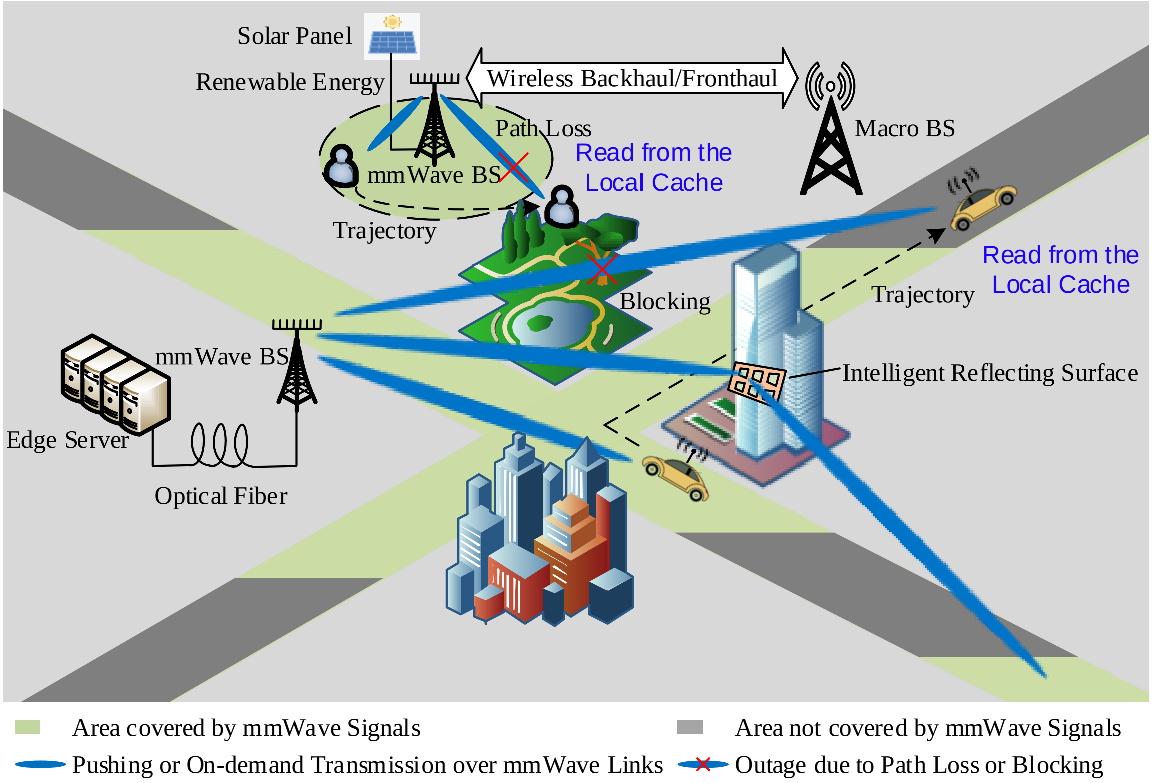

Solar Panel
Renewable Energy
Wireless Backhaul/Fronthaul
Path Loss
mmWave BS
Read from the Local Cache
Macro BS
Trajectory
Blocking
Read from the Local Cache
Trajectory
mmWave BS
Intelligent Reflecting Surface
Edge Server
Optical Fiber
Area covered by mmWave Signals
Area not covered by mmWave Signals
Pushing or On-demand Transmission over mmWave Links
Outage due to Path Loss or Blocking

Figure 2. A hierarchical RAN with partial mmWave coverage, in which caching helps to reduce the required density and deployment costs of mmWave BSs [20].

Caching also allows mmWave BSs to sleep. Therefore, they can be powered by unstable renewable energy. Further, when wireless fronthaul and backhaul are available, the deployment of energy-harvesting-aided mmWave BSs is made more flexible. There is no need to build fixed infrastructures, including a power grid and wired fronthaul and backhaul, for mmWave BSs. Once a site is available, they can be simply placed there. To avoid inter-cell interference, an intelligent spectrum coordinator allocates the spectrum if a nearby mmWave BS is detected. In this case, the deployment of mmWave BSs becomes highly flexible.

2.3. Memory Cost to Be Paid for Caching

A cost to be paid arises from the memories at edge nodes and end devices, which are inexpensive but not free [4]. The memory cost is determined by not only how many bits are cached, but also how long they are cached [21,22].

The average memory size of user devices continues to grow. However, no matter how large a memory is, overflow occurs if the cached contents are never evicted. As a result, a content item should be evicted from the edge node if it becomes unpopular or outdated, or from a user's device if it has been already read by this user. After its eviction, the occupied memory space will be released, which enables memory sharing in the time domain. The eviction time is a challenging issue. Evicting too early may cause possible future requests to be missed, evicting too late wastes the storage resource.

On the other hand, if a file is cached much earlier than its request time, the storage resource is also wasted. If it is cached too late, it may miss user requests, thereby losing the caching gain. In addition, it is inefficient to update memory when the channel remains poor or the spectrum is busy. Hence there is a need to jointly optimize pushing and memory updating, which generalizes the concept of cross-layer design as both wireless links and memories are involved. Such communication-storage coordination becomes very challenging with preference skewness, radio environment dynamics, and coded caching/prefetching.

To attain high memory efficiency, when and how long a content item is cached should be carefully optimized. A memory-efficient eviction policy is presented in [21] to minimize the average memory consumption at user devices. A memory-efficient eviction policy for edge caching is presented in [21] to increase the storage efficiency at BSs. In addition, when the cache memory is full, less popular files can be replaced newly pushed ones [23,24]. Moreover, when a user is experiencing a poor channel condition or the spectrum is crowded, content placement, and memory update are ineffective in terms of transmission costs. Therefore, the physical-layer status should also be considered in the eviction policy. The physical-layer constraints and dynamic environments make the decision policy on the placement time more challenging. Joint optimization of the memory control and content placement emerges as a novel issue of cross-layer scheduling.

In summary, with asynchronous user requests, time-domain memory sharing can be adopted to enable caching with finite memory. To attain high memory efficiency, wireless links and memories should be jointly scheduled from a cross-layer perspective, based on the prediction of what and when a user will request.

3. Request Time Prediction: Beyond Content Popularity

Request time prediction is potentially highly beneficial in proactive caching. Unfortunately, conventional popularity based models, either static or time-varying, are content-specific. They mainly focus on the content popularity distribution among users. For instance, the number of future requests for Facebook video clips can be forecast based on real data. More recently, the prediction accuracy keeps increasing owing to the advances of machine learning and time series analysis. Cross-domain factors such as social connections, recommendation systems, and user activities are also considered in cutting-edge research. Content-specific prediction enables a BS to forecast the distribution of the number of requests arriving in a certain period, based on which the edge caching and multicast pushing can be optimized.

Due to preference skewness and asynchronous requests, content-centric prediction is insufficient for caching at user devices. In particular, content-specific popularity models are incapable of coping with request time prediction. By contrast, a beyond-bit-pipeline RAN not only predicts what will be asked for, but also foresees when it will be requested to unlock the full potential of caching. To this end, time-varying popularity prediction and user-specific prediction play a key role in capturing dynamic popularity and asynchronous requests. We shall discuss how to foresee a user's request time for a certain content item in this section.

3.1. Characterization of Random Request Time

Request time prediction relies on the fact, also observed in [4], that a content item is usually requested by a user at most once. We set a content item's generation time to be the time origin. The item can be requested by a user at a random time after its generation, denoted by X, also referred to as the request delay. If it is never requested by the user, we regard the request delay to be $X = 0^-$. Otherwise, the user will ask for it at $X \geq 0$. The accurate request delay X can hardly be predicted, but its probability density function ($p.d.f.$), denoted by $p(x)$, is predictable. We shall refer to $p(x)$ as the statistical request delay information (RDI), which characterizes our prediction about the request time [11].

RDI provides more knowledge than demand probability and popularity, because we can obtain a user's demand probability α for a content item from its RDI, i.e., $\alpha = \int_0^\infty p(x)dx$. Further, if we assign lower indices i and k to indicate users and content items respectively, the popularity of item k can be characterized by $\frac{N_i}{\sum_k N_k}$, in which $N_k = \sum_i \alpha_{ik}$ represents the expected total number of requests for item k.

RDI is a user-specific prediction. Before the 1990s, computers were expensive and rare, and hence usually shared by a group of users, thereby making individual preference analysis impossible. Today's mobile devices, e.g., smart phones, pads, and laptops, etc. belong to individuals. It is thus possible to learn each individual's preferences by collecting historical requests from their mobile devices. User-specific prediction emerges as a powerful tool to estimate the empirical distribution of the request delay.

3.2. RDI Estimation Methods

Artificial Intelligence (AI) and big data technologies provide powerful tools for understanding user behaviors in the time domain [25–27]. A time-varying popularity prediction for video clips can be found in [28,29], in which real data from YouTube and Facebook are used. In practice, the request time is also affected by one's environments, activities, social connections, etc. For instance, one tends to watch video clips to kill time in the subway or during leisure time, but internet surfing is strictly prohibited while driving. Consequently, user-specific prediction brings together human behavior analysis, natural language processing (NLP), social networks, etc., leading to many cross-disciplinary research opportunities that include but are not limited to

- Learning a user's historical requests and data rating [30,31],
- Exploiting the impact of social networks, recommendation systems, and search engines,
- Discovering relevant content using NLP,
- Analyzing a user behaviors, e.g., activities, mobilities, and localizations.

We demonstrate a method to estimate empirical RDI based on a user's historical requests. In practice, a user is mostly attracted by content labels, titles, or keywords of content items. Content items with the same label may have identical RDI. A BS collects a user's historical request data for N content items with the same label and counts the total number of requests by delay x, denoted by $N(x)$. The cumulative probability density ($c.d.f.$) of the request delay can be estimated as the ratio given by $\hat{P}(x) = \frac{N(x)}{N}$. On the other hand, similar users can be classified according to their preferences. The BS may also collect K similar users' historical requests for a content item and count their total number of requests by delay x, denoted by $K(x)$. Then the empirical $c.d.f.$ of X can be estimated as $\hat{P}(x) = \frac{K(x)}{K}$.

Practical RDI estimation is a rather challenging issue, because the labels of content items are similar but not the same, and so are the request behaviors of users. Moreover, a user's request time can be affected by many other factors, such as social media, relevant reading, and the user's own physical status or environments, as noted above. Besides, one may easily imagine a periodic request fluctuation as users tend to generate more video requests in everyday leisure time. In summary, request time prediction induces many cross-disciplinary research opportunities, bringing together machine learning, recommender systems, social networks, NLP, and human behavior analysis. To protect user privacy, federated learning without the need for uploading raw data is expected to play a key role in capturing popularity skewness and asynchronous requests [32].

4. Fundamental Limits of Caching: A Cross-Layer Perspective

RDI enables cross-layer scheduling of wireless transmission and memory reuse. Their fundamental limits in terms of communications gains and memory costs are thus of interest. Some relevant results are presented in this section.

4.1. Communication Gains

Proactive caching prolongs the transmission time, which enables many possible energy- and/or spectral-efficient physical layer techniques. We are interested in how a content item is pushed given its RDI and what its EE/SE limit is. Quantitative case studies on the EE of pushing over additive white Gaussian noise (AWGN), multiple-input single-output (MISO), and fading channels are presented in [15–17], respectively. A user that tolerates a maximal delay of T seconds may request a content item having B bits. The AWGN channel has a normalized bandwidth and power spectral density of noise. For on-demand transmission, the data rate should be no less than $R = \frac{B}{T}$ (bit/s), and its EE is upper bounded by $\eta_{AWGN} = \frac{R}{2^R - 1}$ (bit/J). When caching is enabled, we shall also refer to the AWGN channel as the pAWGN channel. A decision should be first made whether a content item deserves pushing. To avoid wasting energy, a content item with a demand probability lower than $\frac{R}{2^R+R+\frac{\mathbb{E}\{X\}}{T}+1}$ should not be pushed [15].

For content items that deserve pushing, we should further optimize the transmission powers and rates of the placement and delivery phases. Given the delay constraint T, the more bits that are pushed, the fewer bits that should be delivered after being requested. To maximize the overall EE, we should balance the rates and EEs of the two phases by solving a one-dimensional DC (difference of two convex functions) program [15]. Scaling properties may further reduce the complexity of scheduling in pAWGN channels. Let η^* denote the EE of a pAWGN channel with RDI $p(x)$ and $\alpha = 1$. Then the EE of a pAWGN channel with RDI $\alpha p(x) + (1-\alpha)\delta(x-0^-)$ is lower bounded by $\alpha\eta^*$. Further, a pAWGN channel has the same EE as its normalized version with unit delay constraint, content size R, and scaled RDI $Tp(Tx)$. The lower-left sub-figure of Figure 1 presents η^* of normalized pAWGN channels and enables a table lookup approach to obtain η^*. In practice, lazy scheduling low-order modulations can be adopted in pAWGN channels.

The RDI prediction enlightens more quantitative studies of the cross-layer design of energy- and spectral-efficient pushing with opportunistic scheduling, physical-layer multicasting, OSA, and EH. Although these underlying physical-layer techniques support discontinuous data transmission only, caching provides a virtual real-time experience for users as if they receive their requested data immediately.

4.2. Memory Costs

As noted previously, a cost of caching is increased memory cost, which has to pay the memory cost, which can be reduced by efficiently reusing memory in the time domain. The memory cost is determined by not only how many bits are cached, but also how long they are cached. Memory is wasted if a content item is cached much earlier than being requested or evicted too late after being unpopular. Unfortunately, due to the lack of the request time prediction, how to reuse memory efficiently in the time domain has long been ignored.

A queueing model is formulated in [21] to reveal the limits memory cost and determine whether and how long a content item should be cached based on RDI. Little's Law is adopted to rigorously characterize the relationship between the memory cost and the caching time, i.e., the average memory consumption is the product of the mean caching time and the content arrival rate λ. Given the maximum caching time t, the hit ratio is given by $r = P(t) - P(0)$. Hence, the memory cost and the hit ratio are related as

$$s(r) = \lambda \int_0^{P^{-1}(r+P(0))} x dP(x) + (1-r)P^{-1}(r+P(0)). \tag{1}$$

Equation (1) provides an important insight on the caching policy design, i.e., fixing the maximum caching time is suboptimal when Equation (1) is non-convex. Instead, using different t in a time-sharing manner achieves a smaller memory cost characterized by the lower envelope of Equation (1), as shown in Figure 3. In practice, content items may have different RDIs, with which the maximum caching time for each content class needs optimization. An online algorithm is developed based on the method of Lagrange multipliers to maximize the overall memory efficiency without content placement knowledge. The queueing approach can be also adopted to realize memory-efficient edge caching at BSs [22].

Memory scheduling becomes more challenging in the following three scenarios. First, memory-efficient scheduling with coded caching remains open because the hit ratio of coded caching is still unknown. Second, the hit ratio can be increased by dropping less popular items when the memory is full. This makes the eviction policy more complicated [24]. Finally, the joint scheduling of memories and wireless links generalizes the concept of cross-layer design by involving both the communication and memory units. Deep learning and deep reinforcement learning are expected to play key roles in dealing with the dynamic nature of user requests and radio environments [33–35].

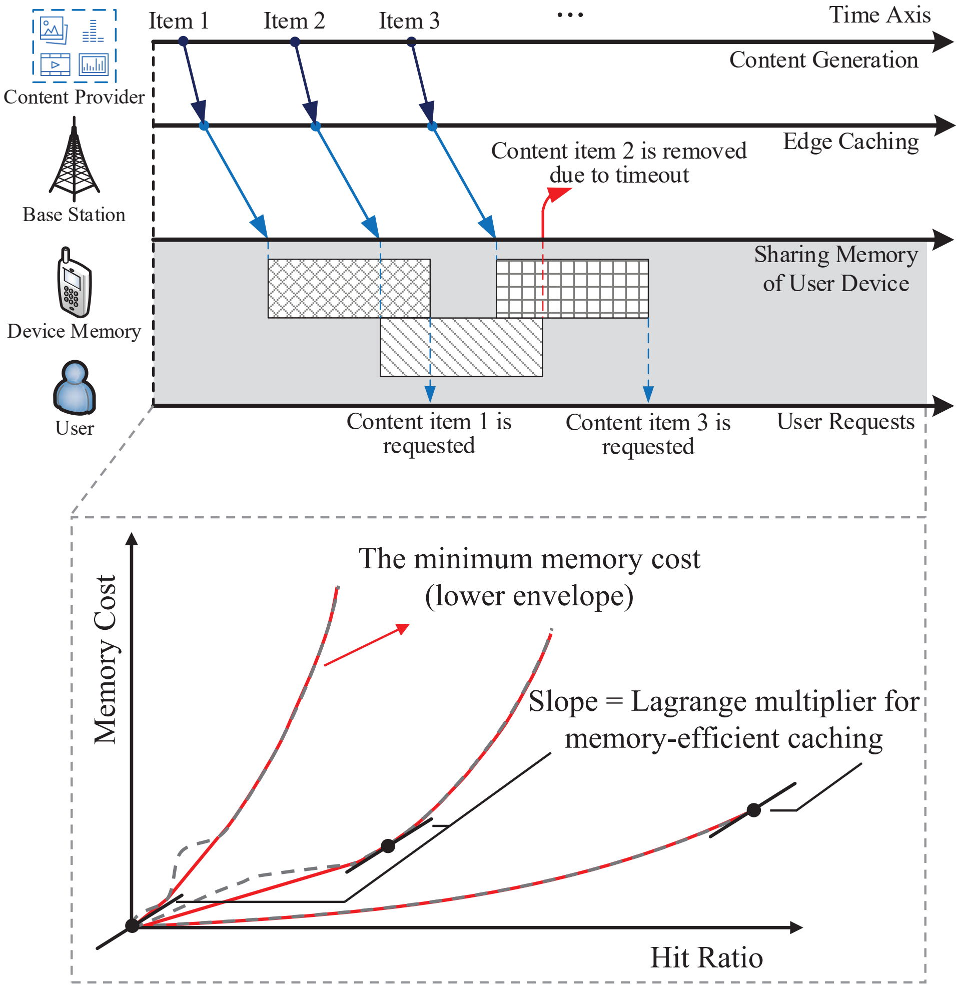

Item 1
Item 2
Item 3
…
Time Axis
Content Provider
Content Generation
Base Station
Edge Caching
Content item 2 is removed due to timeout
Device Memory
Sharing Memory of User Device
User
Content item 1 is requested
Content item 3 is requested
User Requests
Memory Cost
The minimum memory cost (lower envelope)
Slope = Lagrange multiplier for memory-efficient caching
Hit Ratio

Figure 3. A Cross-Layer Perspective on Memory Scheduling: (Upper Sub-figure) Memory Reuse in the Time Domain; (Lower Sub-figure) Memory Cost vs. Hit Ratio [21,22].

5. Pricing: Creating Incentive for Caching

Pricing wireless services has mattered since commercial cellular systems were first deployed. Conventional pricing mechanisms are designed for on-demand transmissions, because today's RANs serve as bit-pipelines only. In this case, a user is charged for her or his access time or data volume received. With caching, however, not all the received data is needed by a user. As no one is willing to pay for what he or she does not need, revolutionary pricing mechanisms are desired in cache-empowered RANs. One straightforward solution is to charge a user for what he or she requests, rather than what he or she receives. In general, pricing should ensure that a user pays a lower service fee, while the RAN's operator has more profit [36]; otherwise no one is willing to adopt caching.

5.1. Pricing Caching Service Using a Hierarchical Architecture

We conceive a hierarchical architecture with virtual network operators (VNOs) [37], as shown in Figure 4. A RAN sells its bandwidth to VNOs, which buy bandwidth to serve their associated users, either by on-demand transmission or caching. If a user cannot find the requested file from the local memory, her or his VNO has to buy bandwidth to serve it. A VNO charges its user for the data volume that the user requests, no matter how a requested file is served.

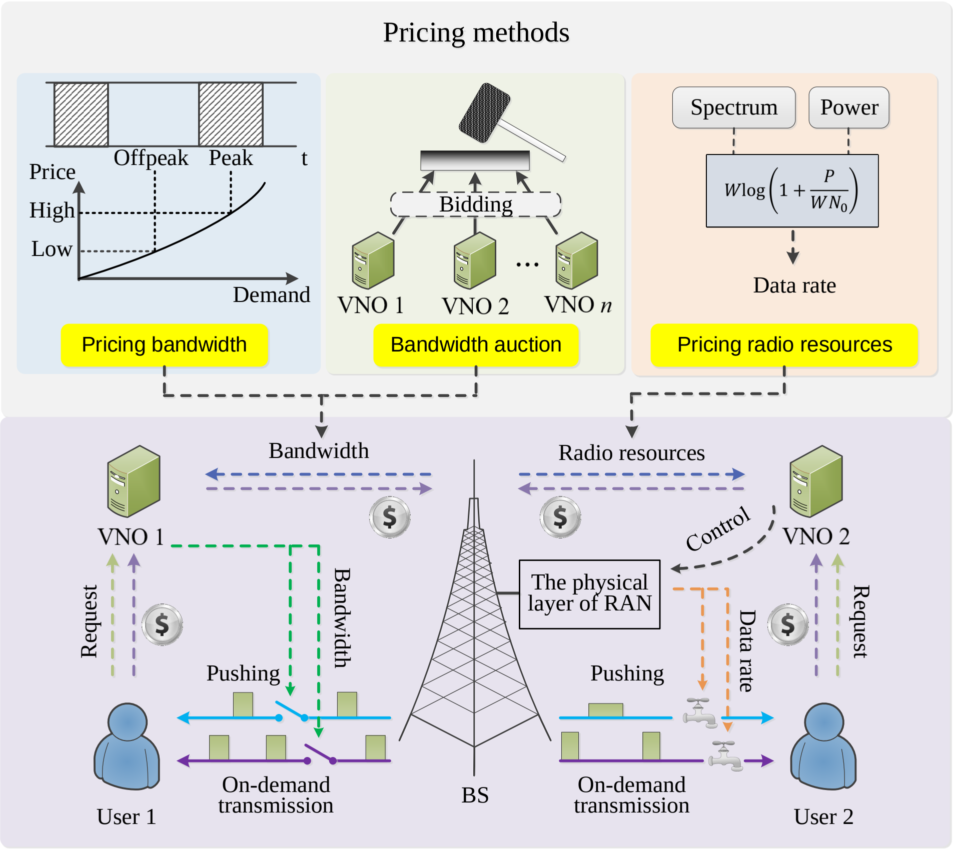


Figure 4. A hierarchical pricing infrastructure for cache-empowered RANs, in which various bandwidth or radio resource pricing mechanisms can be adopted.

A simple scenario in which VNOs schedule bandwidth only is discussed first. The RAN operator charges VNOs a higher bandwidth fee during peak times, because the price is determined by the demand-supply relationship from an economics perspective. If a user's requested file can be found in her or his local cache, the service cost is low. Otherwise, on-demand transmission has to be launched, even if the instantaneous cost is high. On the other hand, caching undesired content wastes the pushing cost. Consequently, VNOs have a strong incentive to maximize the cache hit ratio through accurate request prediction and careful scheduling. This incentive helps to better match the bandwidth demand and supply in the time domain [38]. An alternative way to advocate caching is nonlinear pricing in which the cost per unit spectrum increases with the total amount of spectrum acquired by a user.

Caching should reduce a user's cost for telecommunication services, while increasing the income of spectrum owners and/or RAN operators. The two goals seem to be contradictory, but can be achieved simultaneously due to caching gains. More specifically, the overall service costs are reduced due to the EE and SE gains of caching. Pushing popular items in off-peak time helps to reduce the bandwidth demand during peak times. As such, proactive caching better matches the bandwidth demand and supply in the time domain, which also broadens the cross-disciplinary research of economics and wireless networks.

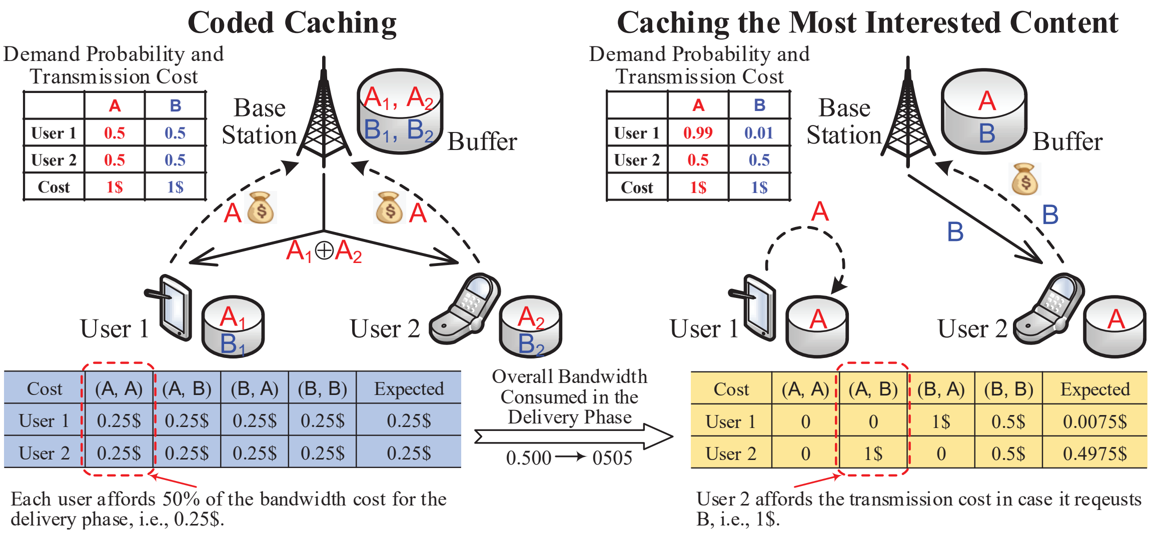
A RAN may adopt an auction that allows VNOs to bid for bandwidth. In this case, the gap between bandwidth prices in peak and off-peak times can become even larger and hence caching saves more cost. Further, if a VNO fails in bidding bandwidth to serve its users, it fails in assuring the QoS, thereby losing users. Therefore, a bandwidth auction may not only increase the income of a RAN, but also eliminate VNOs with weak caching algorithms. To enhance caching policies, VNOs may adopt meta learning or imitation learning. Furthermore, cross-layer pricing mechanisms can be exploited to allow VNOs to control the physical layer directly. Radio resources are then priced dynamically. For instance, idle spectrum and renewable energy are usually cheap or even free. Physical-layer multicasting serves multiple users without additional bandwidth cost. As a result, a VNO can increase its own income by efficiently using radio resources in the physical layer.

5.2. Pricing User Cooperation

Though user cooperation plays a central role in caching, selfish users may be unwilling to cooperate. Pricing is an effective tool to motivate user cooperation in various layers.

Caching-oriented pricing should reward users who contribute more memory for caching or private data for request prediction. A user's hit ratio is increased with more memory used for caching. However, more memory means higher device cost for a user. To reward users contributing more memory for caching, they should enjoy a discount on the telecommunication service fee. On the other hand, the accuracy of request prediction increases with more historical request data or more knowledge about social connections. Sharing these data means more risk in leaking a user's privacy, with which some users are seriously concerned. To gather more data for request prediction, a lower price should be charged for cooperative users. Each VNO may announce its reward policy so that users may choose their favorite VNO according to their own willingness to cooperation in different layers. One powerful tool for developing rewarding mechanisms is a Stackelberg game.

Coded caching holds the potential of scaling the service capability through user cooperation. However, pricing coded caching is rather challenging. As demonstrated in Figure 5, two users U_1 and U_2 may be interested in content items A and B. A natural question is raised as to how the two users should share the bandwidth cost in the delivery phase. If they have equal demand probabilities for A and B, it is fair for each user to pay for half of the cost. In practice, however, popularity skewness exists [4,26]. If U_1 prefers A with a high probability, it will enjoy a high hit ratio by caching A only. Even though the entire B has to be transmitted in the case that U_1 asks for B, the expected bandwidth cost is lower than that of coded caching in which the delivery cost is equally shared. In other words, U_1 is unwilling to participate in coded caching, unless it pays only a small portion of the bandwidth cost. Intuitively, the more demand uncertainty a user has, the more of the cost it should pay. Moreover, a RAN tends to encourage coded caching because the overall bandwidth consumption in the delivery phase can be reduced. A quantitative study on how to price coded caching with popularity skewness can be enabled by game theory [39].


Coded Caching
Caching the Most Interested Content
Demand Probability and Transmission Cost
Base Station
Buffer
A_1, A_2
B_1, B_2
A
A
$A_1 \oplus A_2$
User 1
User 2
A_1
B_1
A_2
B_2
Demand Probability and Transmission Cost
Base Station
Buffer
A
B
A
B
B
User 1
User 2
A
A
Overall Bandwidth Consumed in the Delivery Phase
0.500 → 0505

	A	B
User 1	0.5	0.5
User 2	0.5	0.5
Cost	1$	1$

Cost	(A, A)	(A, B)	(B, A)	(B, B)	Expected
User 1	0.25$	0.25$	0.25$	0.25$	0.25$
User 2	0.25$	0.25$	0.25$	0.25$	0.25$

Each user affords 50% of the bandwidth cost for the delivery phase, i.e., 0.25$.

	A	B
User 1	0.99	0.01
User 2	0.5	0.5
Cost	1$	1$

Cost	(A, A)	(A, B)	(B, A)	(B, B)	Expected
User 1	0	0	1$	0.5$	0.0075$
User 2	0	1$	0	0.5$	0.4975$

User 2 affords the transmission cost in case it reqeusts B, i.e., 1$.

Figure 5. Pricing for Coded Caching: An Effective Throughput Analysis [39].

5.3. Competition and Evolution

Multiple VNOs share the bandwidth provided by a common RAN. Such an infrastructure sharing model results in competition among VNOs, which may bid for bandwidth through auctions. If a VNO fails in getting bandwidth when it has to launch on-demand transmissions, its users will suffer from poor QoS. A VNO providing poor QoS frequently will lose its users. As a result, a VNO has to spend a lot of money to win the bandwidth auction if an on-demand transmission is necessary.

The greater the hit ratio is, the more profit the caching policy brings to a VNO [40]. With more profit, the VNO can afford a higher price to win the bandwidth auction when necessary, thereby assuring QoS. It may also reduce the service fee to attract cost-sensitive users. As a result, VNOs with low cache hit ratios will be either bankrupt due to high service cost or abandoned by users for poor QoS. In other words, the bandwidth auction not only brings more income for a RAN, but also motivates the evolution of prediction and caching policies.

5.4. Pricing Radio Resources, Memory, and Privacy

To fully unlock caching gains, VNOs should be allowed to control the physical layer directly. In particular, a RAN sells its radio resources to VNOs and lets a VNO decide how to use its bought power and spectrum, etc. In this case, VNOs have more freedom and incentive to optimize the SE or EE. A VNO may reduce its energy cost by lazy or opportunistic scheduling, or utilize cheap or even free harvested energy. Physical-layer multicasting can be exploited to push popular files to multiple users without additional spectrum and energy costs.

The memory cost should be considered in pricing. Intuitively, the hit ratio is increased if a user allocates more memory for caching, but more memory means a higher device cost paid by this user. Accordingly, users who are willing to contribute more memory to cache more data should be rewarded e.g., by offering them some discount, as noted previously. Pricing also holds the promise of advocating data sharing that is required by the request prediction. Sharing a user's historical requests or social connections, however, is risky in terms of sacrificing her or his privacy. Being concerned with one's own privacy, a user is usually unwilling to share personal data for request prediction. As a result, users who agree to upload private data should be rewarded. Moreover, serving different users may incur different costs. It is easier to predict the request of a user with low demand uncertainty, which means greater caching gain. Further, users with highly common interests are more likely to be served by multicast pushing. Hence a VNO will tends to reward those users who are interested in commonly popular content.

Sharing the infrastructure, each VNO will announces its own pricing and reward policy. Each user then will chooses her or his favorite VNO based on the willingness of sharing private data, memory allocation, and her or his own preferences. In this context, mechanism design needs more quantitative study from a game-theoretic perspective.

6. Recommendation: Making RANs More Proactive

Classic caching is transparent to users, because they cannot tell whether they are served by on-demand transmission or not. A cache-empowered RAN is made even more proactive if it further shows its users what they might be interested in or even what has been done for them.

6.1. Joint Caching and Recommendation

Recommendation systems (RSs), long recognized as an area of computer science, have been widely used by content providers such as YouTube, Netflix, Tik-Tok, etc. News websites, and even search engines also "recommend" what may interest a user. Nowadays a large portion of data services are driven by RSs. Both RS and caching predict what a user is likely to be interested in, as noted in [4]. A RS aims to a user her or his favorite content items, while a cache-empowered RAN steps further by sending them to the user before being requested. Naturally, joint caching and recommendation (JCR) has attracted some recent attention.

Cache-friendly recommendation is a recent attempt in this area. Its intuitive idea is to push what an RS would recommend and let the user know. By this means, the hit ratio can be improved, as are caching gains. Meanwhile, the RS only recommends what it essentially wants to recommend and avoids showing a user what it is not interested in. In practice, however, the cached files may not be a user's most favorite ones. In this case, cache-friendly recommendation was conceived to recommend content items that are cached but not the most favorable [41]. In addition, recommendation may enhance users' common interests, thereby grouping them to achieve coded-caching or multicasting gain [42]. In both cases, a RAN can enjoy reduced peak-rate and improved SE and/or EE. A cost to be paid is that the user might be less satisfied with the RS if it frequently finds unwanted or useless recommendations. How undesired recommendations harm a user's experience needs more experimental study.

6.2. After-Request Recommendation and Soft Hit

A more adventurous attempt is the recommendation after request, also referred to as the flexible recommendation. Specifically, when a user asks for a content item that has not been cached in the memory, the RS finds some relevant content items from the local cache, if there are any, and recommend them to the user [41]. If the user accepts the recommendation, a soft hit is achieved [43]. Otherwise, on-demand transmission will be launched to satisfy the user. It is a "win-win" solution for both users and RANs because the user enjoys low latency, better QoS, and price reduction by reading from local cache directly, while the RAN enjoys reduced peak-rate and improved SE/EE.

Though low latency sounds attractive, users sometimes need a stronger motivation to accept this "win-win" solution. A potential approach for boosting the soft hit ratio is to reduce or even waive the service fee of the recommended file. Though such a discount reduces a VNO's income from serving the recommended file, it avoids the VNO spending much more money on bidding for peak-time bandwidth. After-request recommendation brings many cross-disciplinary research opportunities. For instance, how to discover relevant content from local cache needs investigation based on NLP or other cross-domain recommendation methods. In turn, the better QoS and lower price provided by caching improve a user's willingness to accept recommendations, leading to an inherent interaction among caching, pricing, and recommendation that remains open.

The joint design of caching, communication, and recommendation substantially broadens the concept of cross-layer design. The "show me the cache" policy makes a RAN much more proactive by affecting user preferences. Instead of simply waiting for a user's command on what and when to transmit, a cache-empowered RAN tells a user what he or she might be interested in and what has been done for him/her. This paradigm-shift architecture motivates a rethinking of effective communications in Figure 6. Classic communication theory cares about whether a message is reliably transmitted, while emerging caching theory cares about whether the message is needed. Then the reliability and utility dimensions form a coordinate system, in which effective communication belongs to the first quadrant. Error detection and correction improve the reliability in the horizontal dimension. Recommendation and request prediction jointly enhance the utility of cached data in the vertical dimension.

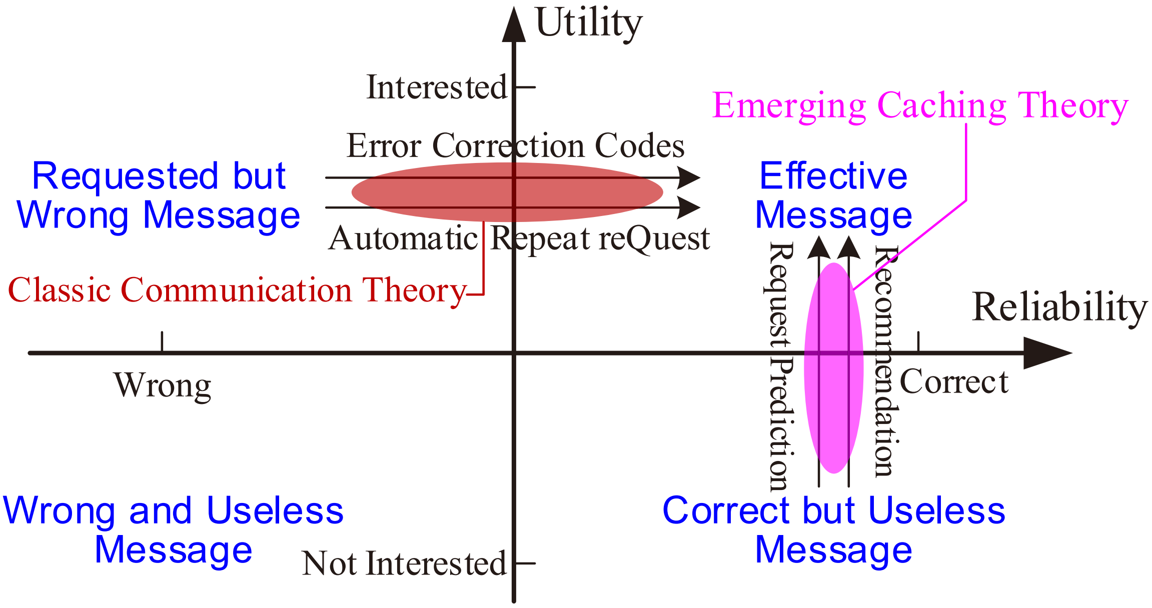


Figure 6. Effective Communications: Reliability vs. Utility.

Cache-friendly recommendation generalizes the research topics on cross-layer designs. First, flexible recommendation might harm a user's QoE if the user is less interested in what is recommended. The Quality-of-Experience (QoE) loss due to the recommendation errors needs more experimental study. How to attain caching gains under QoE constraint emerges as a critical issue. Second, how to apply the cross-domain RS, e.g., based on NLP and social network analysis, raises many cross-disciplinary research opportunities. Third, since a user enjoys low latency and low cost by reading cached content, caching in turn improves a user's willingness to accept recommendations and promotes information propagation in social networks. The interactions between caching, recommendation, and information propagation remain open.

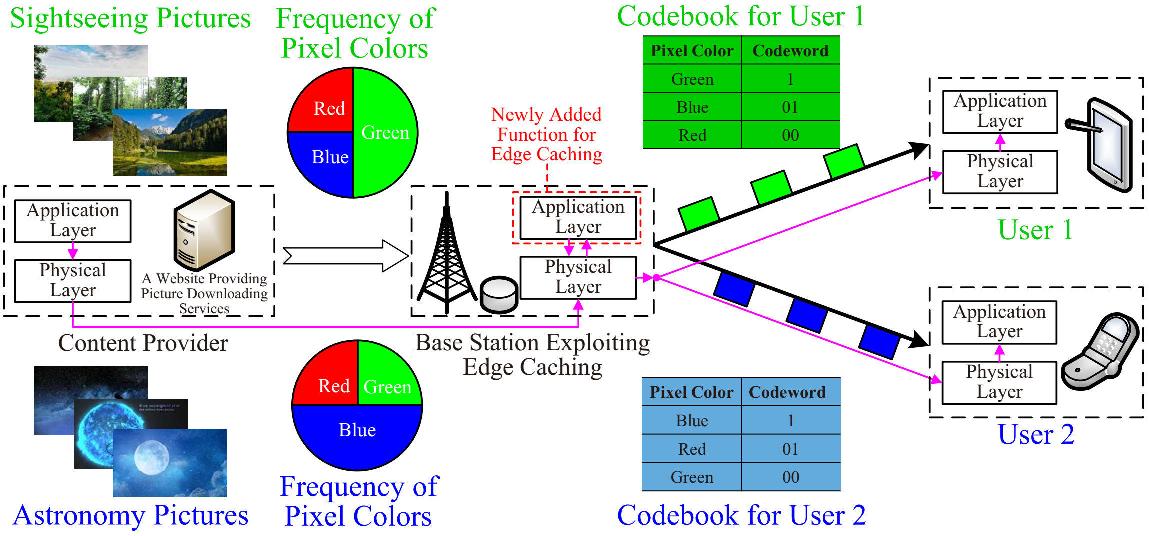
Finally, the "show me the cache" approach allows RANs to affect users. Hence there are potential ethical problems, as users are somehow manipulated. Further, a VNO will tend to advocate common interests for popular files because they help to maximize the multicasting gain in caching. However, such techno-philosophical discussions are beyond the scope of this article.

7. Edge Compression and Proactive Computation in Cache-Empowered RANs

By learning and understanding users, pro-active caching substantially broadens the service types of the application layer. In this section, we review preference-aware compression and proactive edge computing that emerge as beyond-pipeline functions.

7.1. Preference-Aware Compression

One of the key application-layer functions is to compress data. In cache-empowered RANs, user preferences can be exploited to improve the compression ratio of lossless and lossy data compression [44,45], as in Figure 7. A user may prefer a certain subclass of data from a content provider. As a consequence, the symbol distribution of the user's requested data is different to that of data cached at the BS.



Pixel Color	Codeword
Green	1
Blue	01
Red	00

Pixel Color	Codeword
Blue	1
Red	01
Green	00

Figure 7. Edge Compression with User Preference Awareness in Cache-Empowered RANs [44,45].

Consider a toy example with an edge-cached website providing image download services. The pixel colors are assumed to obey a uniform distribution if pixels of all images in the website are counted. Therefore, when the website is transmitted to and cached at the BS, colors are mapped to equal-length codewords for maximizing the compression ratio. Consider a user preferring sightseeing pictures only, in which green pixels appear with a high probability. To reduce the bandwidth consumed in serving this particular user, the BS may re-map green pixels to a shorter codeword. This preference-aware approach is also referred to as edge compression. It will not incur much additional complexity by adopting universal source coding without the knowledge about symbol distributions.

Classic data compression is source-specific, while edge compression is user-specific. It challenges the current protocol stack inspired by the source-channel separation theorem. More particularly, no application-layer function is performed at present-day BSs. By contrast, edge compression requires BSs to implement application-layer protocols. Other research opportunities include edge-compression-friendly recommendation and multicasting. RSs may change a user's preferences, thereby holding the promise of further improving the compression ratio at the edge. The existing physical-layer multicasting relies on using a common codebook of source coding, which is not compatible with user-specific compression. Further cross-layer design of edge compression and multicasting is desired.

7.2. When Caching Meets MEC

Both caching and RSs rely on request predictions, resulting in much computational load. Mobile edge computing holds the promise of providing the required computational power. MEC is expected to merge with caching in the era of 6G [3] in order to predict user requests, make caching decisions, price caching services, and realize cache-friendly recommendation and preference-aware compression, while assuring low latency and privacy protection. Similar to RANs, peak-time overload and congestion exist in MEC. Therefore users may suffer from a large latency when a MEC unit is heavily loaded [46].

To reduce the computational workload during peak times, we borrow the idea of caching based on an observation that the computational tasks required by users are also predictable. By predicting users' most common computation tasks in the future and based on the pro-actively cached data, off-peak-time computational resources can be fully exploited to implement these tasks before user requests and hence reduce the service delay. For instance, once a content item becomes popular over the Internet or on social networks, joint caching and recommendation can be triggered to foresee if the content is worth recommending or pushing to a user. In this case, the computational task is implemented without the user's order to reduce the MEC's peak load.

Proactive MEC is made feasible with cached data. Such an idea can be further borrowed by other MEC applications, e.g., mobile navigation or autonomous driving, in which route planning between a user's favorite locations can be determined based on the traffic information pushed. The concept of recommendation can also be generalized given increasing MEC capabilities. Not only what to read in cyberspace, but also how to enhance the physical layer performance, can be suggested. For instance, an RS can suggest that a user moves to a nearby mmWave BS if the user asks for a large file that has not been cached. If the user agrees to do so, navigation software will guide it to the selected mmWave-covered cell. To realize the above beyond-pipeline functions, computation task prediction and joint scheduling of communication, caching, and computation emerge as open issues for making MEC more proactive.

8. Conclusions

This article, as a humble attempt to envision wireless caching from a cross-layer perspective, has described a vision of caching that makes RANs more than bit-pipelines. A cache-empowered RAN determines by itself what, when, and how to transmit and cache without waiting for requests from users. The prediction of user requests in the time domain enables efficient joint scheduling of wireless links and memories. To motivate the application and evaluation of caching, we have conceived a novel hierarchical pricing architecture that advocates caching and user cooperation. We have also discussed joint caching and recommendation to serve users in a more proactive way. Furthermore, the user-specific prediction also motivates edge compression and proactive MEC as new applications in the beyond-pipeline RANs. Since caching holds the promise of making RANs more than bit-pipelines, it will generalize the concept of cross-layer design to be multi-unit collaboration of communication, computation, recommendation, and memory units. Many cross-disciplinary research opportunities emerge in considering beyond-bit-pipeline RANs.

Author Contributions: Conceptualization, W.C. and H.V.P.; methodology, W.C. and H.V.P.; validation, W.C. and H.V.P.; formal analysis, W.C. and H.V.P.; writing—original draft preparation, W.C. and H.V.P.; writing—review and editing, H.V.P.; visualization, W.C.; project administration, W.C. and H.V.P.; funding acquisition, W.C. and H.V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the U.S. National Science Foundation under Grant CCF-1908308 and the National Natural Science Foundation of China under Grant No. 61671269.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cisco. Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper, 2019. Available online: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf (accessed on 17 May 2020).
2. Rappaport, T.S.; Xing, Y.; Kanhere, O.; Ju, S.; Madanayake, A.; Mandal, S.; Alkhateeb, A.; Trichopoulos, G.C. Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond. *IEEE Access* **2019**, *7*, 78729–78757. [CrossRef]
3. Letaief, K.B.; Chen, W.; Shi, Y.; Zhang, J.; Zhang, Y.-J.A. The roadmap to 6G–AI empowered wireless networks. *IEEE Commun. Mag.* **2019**, *57*, 84–90. [CrossRef]
4. Paschos, G.; Baştuğ, E.; Land, I.; Caire, G.; Debbah, M. Wireless caching: Technical misconceptions and business barriers. *IEEE Commun. Mag.* **2016**, *54*, 16–22. [CrossRef]

5. Paschos, G.S.; Iosifidis, G.; Tao, M.; Towsley, D.; Caire, G. The role of caching in future communication systems and networks. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1111–1125. [CrossRef]
6. Bai, B.; Wang, L.; Han, Z.; Chen, W.; Svensson, T. Caching based socially-aware D2D communications in wireless content delivery networks: A hypergraph framework. *IEEE Wirel. Commun.***2016**, *23*, 74–81. [CrossRef]
7. Wang, L.; Wu, H.; Ding, Y.; Chen, W.; Poor, H.V. Hypergraph-based wireless distributed storage optimization for cellular D2D underlays. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 2650–2666. [CrossRef]
8. Yan, Q.; Chen, W.; Poor, H.V. Big data driven wireless communications: A human-in-the-loop pushing technique for 5G systems. *IEEE Wirel. Commun.* **2018**, *25*, 64–69. [CrossRef]
9. Maddah-Ali, M.A.; Niesen, U. Fundamental limits of caching. *IEEE Trans. Inf. Theory* **2014**, *60*, 2856–2867. [CrossRef]
10. Maddah-Ali, M.A.; Niesen, U. Cache-aided interference channels. *IEEE Trans. Inf. Theory* **2019**, *65*, 1714–1724. [CrossRef]
11. Chen, W.; Poor, H.V. Content pushing with request delay information. *IEEE Trans. Commun.* **2017**, *65*, 1146–1161. [CrossRef]
12. Lu, Y.; Chen, W.; Poor, H.V. Multicast pushing with content request delay information. *IEEE Trans. Commun.* **2018**, *66*, 1078–1092. [CrossRef]
13. Lu, Y.; Chen, W.; Poor, H.V. Coded joint pushing and caching with asynchronous user requests. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1843–1856. [CrossRef]
14. Lu, Y.; Chen, W.; Poor, H.V. A unified framework for caching in arbitrary networks. In Proceedings of the 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 19–21 November 2018; pp. 1–5.
15. Huang, W.; Chen, W.; Poor, H.V. Energy efficient pushing in AWGN channels based on content request delay information. *IEEE Trans. Commun.* **2018**, *66*, 3667–3682. [CrossRef]
16. Lin, Z.; Chen, W. Content pushing over multiuser MISO downlinks with multicast beamforming and recommendation: A cross-layer approach. *IEEE Trans. Commun.* **2019**, *67*, 7263–7276. [CrossRef]
17. Xie, Z.; Lin, Z.; Chen, W. Power and rate adaptive pushing over fading channels. *IEEE Trans. Wirel. Commun.* **2021**, early access. [CrossRef]
18. Li, C.; Chen, W. Content pushing over idle timeslots: Performance analysis and caching gains. *IEEE Trans. Wirel. Commun.* **2021**, early access. [CrossRef]
19. Zhou, S.; Gong, J.; Zhou, Z.; Chen, W.; Niu, Z. GreenDelivery: Proactive content caching and push with energy-harvesting-based small cells. *IEEE Commun. Mag.* **2015**, *53*, 142–149. [CrossRef]
20. Xie, Z.; Chen, W.; Poor, H.V. Exploiting millimeter wave hotspots in two-tier heterogeneous networks with mobility-enabled pushing. *IEEE Glob. Commun. Conf. (Globecom)* **2021**, submitted.
21. Chen, W.; Poor, H.V. Caching with time domain buffer sharing. *IEEE Trans. Commun.* **2019**, *67*, 2730–2745. [CrossRef]
22. Xie, Z.; Chen, W. Storage-efficient edge caching with asynchronous user requests. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 229–241. [CrossRef]
23. Gao, J.; Zhang, S.; Zhao, L.; Shen, X. The design of dynamic probabilistic caching with time-varying content popularity. *IEEE Trans. Mob. Comput.* **2021**, *20*, 1672–1684. [CrossRef]
24. Hui, H.; Chen, W.; Wang, L. Caching with finite buffer and request delay information: A markov decision process approach. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 5148–5161. [CrossRef]
25. Bharath, B.N.; Nagananda, K.G.; Gündxuxz, D.; Poor, H.V. Caching with time-varying popularity profiles: A learning-theoretic perspective. *IEEE Trans. Communn.* **2018**, *66*, 3837–3847. [CrossRef]
26. Lee, M.-C.; Molisch, A.F.; Sastry, N.; Raman, A. Individual preference probability modeling and parameterization for video content in wireless caching networks. *IEEE/ACM Trans. Netw.* **2019**, *27*, 676–690. [CrossRef]
27. Yang, L.; Guo, X.; Wang, H.; Chen, W. A video popularity prediction scheme with attention-based LSTM and feature embedding. In Proceedings of the 2020 IEEE Global Communications Conference (GLOBECOM 2020), Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
28. Yang, P.; Zhang, N.; Zhang, S.; Yu, L.; Zhang, J.; Shen, X. Content Popularity Prediction Towards Location-Aware Mobile Edge Caching. *IEEE Trans. Multimed.* **2019**, *21*, 915–929. [CrossRef]
29. Tang, L.; Huang, Q.; Puntambekar, A.; Vigfusson, Y.; Lloyd, W.; Li, K. Popularity prediction of facebook videos for higher quality streaming. In Proceedings of the USENIX Annual Technical Conference (USENIX ATC), Santa Clara, CA, USA, 12–14 July 2017; pp. 111–123.
30. Wu, J.; Yang, C.; Chen, B. Proactive caching and bandwidth allocation in heterogeneous network by learning from historical number of requests. *IEEE Trans. Commun.* **2020**, *68*, 4394–4410. [CrossRef]
31. Cheng, P.; Ma, C.; Ding, M.; Hu, Y.; Lin, Z.; Li, Y.; Vucetic, B. Localized small cell caching: A machine learning approach based on rating data. *IEEE Trans. Commun.* **2019**, *67*, 1663–1676. [CrossRef]
32. Yang, K.; Shi, Y.; Zhou, Y.; Yang, Z.; Fu, L.; Chen, W. Federated machine learning for intelligent IoT via reconfigurable intelligent surface. *IEEE Netw.* **2020**, *34*, 16–22. [CrossRef]
33. Li, L.; Xu, Y.; Yin, J.; Liang, W.; Li, X.; Chen, W.; Han, Z. Deep reinforcement learning approaches for content caching in cache-enabled D2D networks. *IEEE Internet Things J.* **2020**, *7*, 544–557. [CrossRef]
34. Li, L.; Cheng, Q.; Tang, X.; Bai, T.; Chen, W.; Ding, Z.; Han, Z. Resource allocation for NOMA-MEC systems in ultra-dense networks: A learning aided mean-field game approach. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 1487–1500. [CrossRef]

35. Chen, Q.; Wang, W.; Chen, W.; Yu, F.R.; Zhang, Z. Cache-enabled multicast content pushing with structured deep learning. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2135–2149. [CrossRef]
36. Liu, L.; Zhou, Y.; Yuan, J.; Zhuang, W.; Wang, Y. Economically optimal MS association for multimedia content delivery in cache-enabled heterogeneous cloud radio access networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1584–1593. [CrossRef]
37. Huang, W.; Chen, W.; Poor, H.V. Request delay-based pricing for proactive caching: A stackelberg game approach. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2903–2918. [CrossRef]
38. Lin, Z.; Huang, W.; Chen, W. Bandwidth and storage efficient caching based on dynamic programming and reinforcement learning. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 206–209. [CrossRef]
39. Lu, Y.; Li, C.; Chen, W.; Poor, H.V. On the effective throughput of coded caching with heterogeneous user preferences: A game theoretic perspective. *IEEE Trans. Commun.* **2021**, *69*, 1387–1402. [CrossRef]
40. Hui, H.; Chen, W. A pricing-based joint scheduling of pushing and on-demand transmission over shared spectrum. In Proceedings of the 2020 IEEE Global Communications Conference (GLOBECOM 2020), Taipei, Taiwan, 7–11 December 2020; pp. 1–5. [CrossRef]
41. Chatzieleftheriou, L.E.; Karaliopoulos, M.; Koutsopoulos, I. Jointly optimizing content caching and recommendations in small cell networks. *IEEE Trans. Mob. Comput.* **2019**, *18*, 125–138. [CrossRef]
42. Zhu, B.; Chen, W. Coded caching with moderate recommendation: Balancing delivery rate and quality of experience. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 1456–1459. [CrossRef]
43. Sermpezis, P.; Giannakas, T.; Spyropoulos, T.; Vigneri, L. Soft cache hits: Improving performance through recommendation and delivery of related content. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1300–1313. [CrossRef]
44. Lu, Y.; Chen, W.; Poor, H.V. User preference aware lossless data compression at the edge. *IEEE Trans. Commun.* **2020**, *68*, 3792–3807. [CrossRef]
45. Lu, Y.; Chen, W.; Poor, H.V.; User preference aware lossy data compression for edge caching. In Proceedings of the 2020 IEEE Global Communications Conference (GLOBECOM 2020), Taipei, Taiwan, 7–11 December 2020; pp. 1–5. [CrossRef]
46. Han, D.; Chen, W.; Bai, B.; Fang, Y. Offloading optimization and bottleneck analysis for mobile cloud computing. *IEEE Trans. Commun.* **2019**, *67*, 6153–6167. [CrossRef]