Age of Information and Delay Tradeoff with Freshness-Aware Mobile Edge Cache Update

Shan Zhang*, Liudi Wang*, Hongbin Luo*, Xiao Ma†, and Sheng Zhou‡
Email: zhangshan18@buaa.edu.cn, wldouc@163.com, luohb@buaa.edu.cn,
maxiao18@bupt.edu.cn, sheng.zhou@tsinghua.edu.cn
*School of Computer Science and Engineering, Beihang University, Beijing, 100083, China
†Beijing University of Posts and Telecommunications, Beijing, 100876, China
‡Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

***Abstract*—Mobile edge caching is an effective way to reduce the service delay of content delivery, where the popular contents can be pro-actively stored in proximity to users. In practice, the cached contents should be updated timely to avoid information staleness, in case that the information of a content changes with time and environment. However, cache update consumes additional transmission resources, which can degrade the delay performance. This work studies the fundamental tradeoff relationship between the content freshness (depicted by the age of information (AoI)) and service delay in mobile edge caching networks, and proposes a freshness-aware cache update scheme to achieve the AoI-delay balance. In specific, the base station will fetch the latest version of a content before delivery, if the AoI is larger than a certain threshold (i.e., update window size). The average AoI and service delay are derived in closed forms through approximated analysis of queueing systems, revealing a tradeoff relationship with respect to the update window size. Extensive simulations are conducted on the OMNeT++ platform, which validates the analytical results. Both the analytical and simulation results show that the proposed scheme can flexibly balance the average AoI and delay on demand, by tuning the update window size. Furthermore, the proposed scheme can also avoid frequent update in case of heavy content requests, whereby the AoI and delay are regulated by setting the appropriate update window size.**

***Index Terms*—mobile edge caching, age of information (AoI), service delay, cache update, content dynamics**

I. Introduction

Mobile edge caching has been proposed as a promising network paradigm to provide content services based on request information, where contents can be stored at base stations (BSs) or even end devices in addition to remote data centers [1]. As such, mobile users can be served in proximity, which can effectively reduce the end-to-end delay of content services [2], [3]. However, the information of a content may change with time and environment in practice (i.e., the time-varying contents), such as the content of the same URL, the traffic condition of a road segment, the availability of a parking lot, and the real-time street map. Accordingly, the cached contents should be updated to the newest versions timely, otherwise mobile users may receive staled or even invalid information, severely degrading the quality of experience. Thus, content freshness is now considered as important performance in addition to service delay, which has been receiving increasing attention in different application scenarios like cloud games, mobile edge caching and computing [4]–[6]. To characterize freshness of time-varying contents, the age of information (AoI) has been proposed, which is defined as the time elapsed since the generation of the content [7].

In mobile edge caching networks, guaranteeing content freshness and service delay may be inconsistent due to the constrained transmission resources, posing great challenges to cache update. Frequent cache update helps to improve content freshness, but consumes additional transmission resources of BSs and degrades the service delay. Therefore, effective cache update schemes are demanded to balance the content freshness and service delay, which has not been well studied in existing literature. The existing studies on cache update mainly focus on the variation of content popularity while assuming static contents, i.e., the information of a content remains unchanged during life time. However, in fact, cache update is more critical for the time-varying contents, as users expect effective and fresh information from networks.

In this work, we revisit the cache update problem considering the time-varying contents. The fundamental tradeoff relationship between content freshness and service delay is investigated in mobile edge caching networks, whereby a freshness-aware cache update scheme is proposed to achieve the optimal balance. A cache-enabled BS works as a sink node, which collects the time-varying contents published by surrounding source nodes and serves mobile users on demand. To ensure the content freshness, the BS always checks the AoI before delivering the content to a mobile user. If the AoI is larger than a certain threshold (defined as *update window size*), the BS will fetch the latest version from the source node and send to the mobile user. Meanwhile, the BS will also update the cache along with this process. The content update and delivery process is analyzed by applying the queueing theory, whereby the approximated average AoI and service delay are derived in closed forms. In specific, the average AoI increases with the update window size in a convex manner, and shows asymptotic linear relationship as the update window size is sufficiently large. In contrast, the average delay decreases with the update window size in a

This work was supported in part by the Nature Science Foundation of China No. 61801011, 91638204, in part by the Fundamental Research Funds for the Central Universities, China, under Grant No. YWF-18-BJ-J-61, JZX7Y20190251030001.

978-1-7281-0962-6/19/$31.00 ©2019 IEEE

concave manner, and finally levels off for large update window size. Thus, the analytical results reveal the tradeoff relationship between AoI and delay, with respect to the update window size. Extensive simulations are conducted on the OMNeT++ platform, validating the above analytical results. Furthermore, the results show that the proposed scheme restrains the update probability in case of high request rates, which can be considered as a regulation of cache update.

The remaining of this paper is organized as follows. Section II reviews the existing works on cache update, Section III proposes the freshness-aware cache update scheme, Section IV conducts theoretical analysis on the average AoI and delay performance, Section V shows simulation results, and Section VI finally draws conclusions.

II. Literature Review

In most of the existing literature, the cached contents are updated according to the variation of popularity, aiming at maximizing the content hit rate. The cache time of each content has been optimized based on the prior knowledge of popularity dynamics to minimize the cache costs [8]. In case of unknown content popularity, adaptive content update schemes can be applied, where the least used contents are moved out of cache [9]. Furthermore, recent works have utilized machine learning approaches in mobile caching, whereby online cache update schemes have been devised by learning and predicting the content popularity [10]. Note that these existing works mainly conduct cache update according to the variation of content popularity, while the dynamic variation of content information has not been taken into account.

The very recent works in [6], [11] have introduced content freshness in mobile edge caching networks, which are most related to this work. Considering that the popularity of a content can fade with time, Kam *et al.* have proposed to remove the contents with higher AoI for cache update [6]. However, this work still focuses on the popularity variation instead of the content dynamics. The content dynamics is considered in [11], and a content update scheme is proposed to minimize the average AoI of a local cache system. Compared with [11], this work goes one step further by jointly optimizing the average AoI and delay, such that users can obtain the fresh contents within short delay. To this end, a freshness-aware cache update scheme is devised, considering the coupling effect of content update and delivery due to the transmission resource constraints.

III. Mobile Edge Caching System Model

A typical mobile edge caching network is considered, as shown Fig. 1. Source nodes, which monitor the surrounding environment and publish contents, are randomly distributed within the coverage of a BS. The contents can refer to the status of traffic jams, the availability of a parking lot, and the real-time 3D map of a street. Note that these contents may change with time, and thus the source nodes keep publishing new versions of the contents to reflect the timely status. The BS is cache-enabled and works as a sink node, which collects the published contents to serve mobile users on demand.

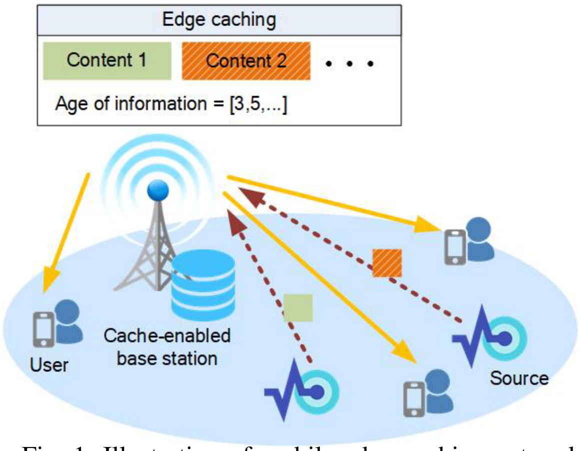


Fig. 1: Illustration of mobile edge caching networks.

The proposed freshness-aware cache update scheme at the BS is as follows. Suppose that the cache size is sufficiently large and all contents are cached at the BS. For each cached content, the BS records the AoI to monitor the freshness, and sets a cache update window, denoted by W, based on the AoI requirement. Before delivering a content upon the user request, the BS will first check the AoI and determine whether to update the content or not. If the AoI is smaller than W, the BS will deliver the cached content directly. Otherwise, the BS will fetch the latest version from the source node and then deliver to the mobile user. Meanwhile, the cached content will be also updated to the newest version and the corresponding AoI will be reset.

The transmission resource of the BS is split into orthogonal channels, and each channel is utilized for the content update and delivery of one source node. We focus on the service of one source node to conduct analysis, since the service processes of different source nodes are independent. Denote by λ the request arrival rate, and assume mobile users raise requests randomly following a Poisson process. The BS serves content requests in a First-In-First-Out (FIFO) manner through wireless unicast, for both the uplink content fetching and the downlink content delivery. Denote by T_{U} and T_{D} the transmission time of cache update (i.e., from the source node to the BS) and content delivery (i.e., from the BS to the mobile users), respectively. In practice, T_{U} and T_{D} are both uncertain, depending on the transmission distance, random channel fading and interference. To conduct theoretical analysis, T_{U} and T_{D} are modeled to follow exponential distributions with parameters of μ_{U} and μ_{D}, respectively. Accordingly, μ_{U} and μ_{D} can be interpreted as the content update and delivery rate, receptively.

Notice that the update window size W can guarantee the content freshness of user received contents. In specific, users will receive fresher contents by setting smaller W. However, the cached content with smaller window size W has to be updated more frequently, which will introduce more transmission loads in the uplink, increasing the average service delay of BS. Thus, there exists a tradeoff relationship between the AoI and service delay, and the update window size should

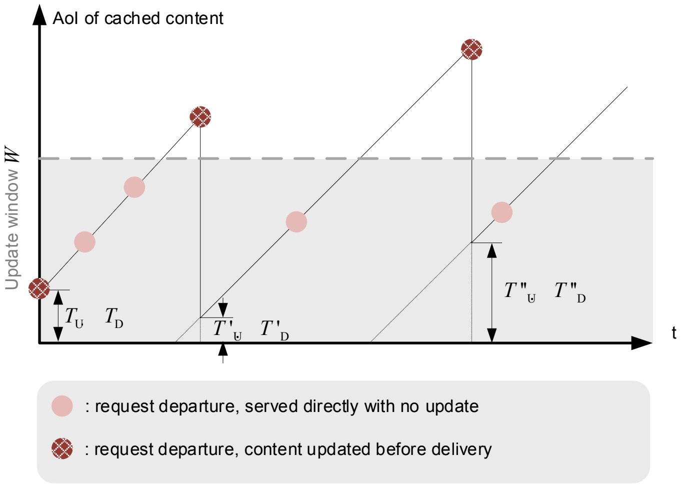

AoI of cached content
Update window W
T_U T_D
T'_U T'_D
T''_U T''_D
t
: request departure, served directly with no update
: request departure, content updated before delivery

Fig. 2: The request departure process with AoI variation illustration.

be carefully designed to achieve the balance. To this end, we analyze the AoI and service delay with respect to the update window size in the next section.

IV. AoI and Service Delay Analysis

This section conducts analysis on the BS service process in an approximated manner, whereby the average AoI and service delay are obtained in closed forms.

A. *Service Process Analysis*

As the content request arrival is a Poisson process, the service process of the BS can be modeled as a FIFO M/G/1 queue. Denote by X the service time of a content request. X depends on whether the request triggers cache update or not:

$$X = \begin{cases} T_\text{D}, & \text{directly delived,} \\ T_\text{D} + T_\text{U}, & \text{cache updated.} \end{cases} \tag{1}$$

Denote by p the probability that a request triggers cache update. Thus, the average service time is given by

$$\mathbb{E}[X] = \frac{p}{\mu_\text{U}} + \frac{1}{\mu_\text{D}}. \tag{2}$$

Notice that the average service time increases with the cache update probability p, while p further depends on the cache update window size W, the arrival and departure processes of content requests. Therefore, the key issue is to obtain the update probability p by analyzing the service process. However, the accurate form of p cannot be derived since the departure process of a M/G/1 queue has time correlations. To provide insights on update window management, we simplify the performance analysis and approximate the M/G/1 queue with a M/M/1 queue whose arrival rate is set to λ and service rate is set to $1/\mathbb{E}[X]$.

B. *Content Update Probability*

The departure process of content requests is shown as Fig. 2. Each circle denotes a event that a request is served and leaves the queueing system. The solid circles correspond to the requests which are directly served without cache update, while the shadowed ones represent the requests which trigger cache update. Suppose one request which triggers cache update finishes service at time zero, without losing generality. Accordingly, the AoI of cached content is reset to the $T_\text{U}+T_\text{D}$ at time zero, i.e., the two-hop transmission time of content fetching and delivery. Then, the AoI increases as time goes until another content request triggers update.

Note that the requests departing during time $[0, W - T_\text{U} - T_\text{D}]$ cannot trigger cache update, as the corresponding AoI is below the update window size. According to the properties of M/M/1 queuing, the departure process also follows Poisson process of rate λ in the stable state. Thus, the time period between two successive request departure follows exponential distribution of parameter λ. Consider the time period between two content updates as a update cycle, wherein N requests are served directly without cache update. The update cycle length is random, depending on the arrival time of next request triggering cache update. However, the N requests directly served should all happen within the window under the proposed update scheme. Accordingly, N follows Poisson distribution of mean $(W - T_\text{U} - T_\text{D})\lambda$. By approximating the random transmission time by the average value, the update probability can be obtained:

$$\begin{aligned} p = \mathbb{E}\left[\frac{1}{N+1}\right] &= \sum_{n=0}^{\infty} \frac{1}{n+1} \frac{\bar{N}^{-n}}{n!} e^{-\bar{N}} \\ &= \frac{1}{\bar{N}} \sum_{n=1}^{\infty} \frac{\bar{N}^n}{-n!} e^{-\bar{N}} = \frac{1-e^{-\bar{N}}}{\bar{N}}, \end{aligned} \tag{3}$$

where $\bar{N} = (W - \frac{1}{\mu_\text{U}} - \frac{1}{\mu_\text{D}})\lambda$, reflecting average number of requests directly served in each update cycle. Therefore, the cache update probability p can be obtained:

$$p = \frac{1}{(W - \frac{1}{\mu_\text{U}} - \frac{1}{\mu_\text{D}})\lambda} \left[1 - e^{-(W - \frac{1}{\mu_\text{U}} - \frac{1}{\mu_\text{D}})\lambda}\right]. \tag{4}$$

Note that the update probability decreases convexly with the traffic load λ and the update window size W. In specific, take the first- and second-order derivatives of p with respect to $\bar{N}$, and we have

$$\frac{\partial p}{\partial \bar{N}} = -\frac{1-e^{-\bar{N}}}{\bar{N}^2} + \frac{e^{-\bar{N}}}{\bar{N}} = \frac{e^{-\bar{N}}}{\bar{N}^2}\left(\bar{N} + 1 - e^{\bar{N}}\right), \tag{5}$$

and

$$\frac{\partial^2 p}{\partial \bar{N}^2} = \frac{2e^{-\bar{N}}}{\bar{N}^3}\left(e^{\bar{N}} - 1 - \bar{N} - \frac{\bar{N}^2}{2}\right). \tag{6}$$

Furthermore, $e^x \geq 1 + x + \frac{1}{2}x^2 \; \forall x \geq 0$, based on the Taylor's series of e^x. Accordingly, $\frac{\partial p}{\partial \bar{N}} \leq 0$ and $\frac{\partial^2 p}{\partial \bar{N}^2} \geq 0$. Thus, p is a decreasing convex function with respect to $\bar{N}$. As $\bar{N}$ increases with λ and W linearly, p also decreases convexly with λ and W, respectively.

The important insight of these analytical results is that the proposed freshness-aware update scheme restrains the update frequency of frequently requested contents (i.e., the popular contents), as the update probability p decreases with λ. As such, the proposed method can regulate the update frequency under different traffic loads, whereby the AoI and delay performance can be controlled by setting appropriate update window size.

C. Age of Information Analysis

Based on the derived update probability, the average AoI can be derived. In specific, we consider one update cycle, during which the BS completes $N+1$ content requests. Among the $N+1$ requests, the first one triggers cache update, and thus the AoI of user received content equals to $T_\mathrm{U}+T_\mathrm{D}$. The departure time of the other N requests are uniformly distributed in the range of $[T_\mathrm{U}+T_\mathrm{D}, W]$, according to the properties of M/M/1 queueing systems. The average AoI of user received contents is thus obtained:

$$\begin{aligned}\bar{A} &= \underset{\{N,T_\mathrm{U},T_\mathrm{D}\}}{\mathbb{E}}\left[\frac{1}{N+1}\left(T_\mathrm{U}+T_\mathrm{D}+N\frac{T_\mathrm{U}+T_\mathrm{D}+W}{2}\right)\right] \\ &= \underset{\{N,T_\mathrm{U},T_\mathrm{D}\}}{\mathbb{E}}\left[\frac{W+T_\mathrm{U}+T_\mathrm{D}}{2} - \frac{W-T_\mathrm{U}-T_\mathrm{D}}{2(N+1)}\right] \\ &= \frac{1}{2}\left(W+\frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}\right)-\frac{1}{2}\left(W-\frac{1}{\mu_\mathrm{U}}-\frac{1}{\mu_\mathrm{D}}\right)\underset{\{N\}}{\mathbb{E}}\left[\frac{1}{N+1}\right] \\ &= \frac{1}{2}\left(W+\frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}\right)-\frac{1}{2\lambda}\left(1-e^{-\left(W-\frac{1}{\mu_\mathrm{U}}-\frac{1}{\mu_\mathrm{D}}\right)\lambda}\right).\end{aligned} \tag{7}$$

The derived average AoI in Eq. (7) reveals the influence of the update window size on content freshness. In specific, the average AoI $\bar{A}$ increases with the update window size W in a convex manner, and has a asymptotically linear relationship when the update window size is large. By taking the first- and second-order derivatives of $\bar{A}$, we have

$$\begin{aligned}\frac{\partial\bar{A}}{\partial W} &= \frac{1}{2}-\frac{1}{2\lambda}e^{-\left(W-\frac{1}{\mu_\mathrm{U}}-\frac{1}{\mu_\mathrm{D}}\right)\lambda}\lambda \\ &= \frac{1}{2}\left[1-e^{-\left(W-\frac{1}{\mu_\mathrm{U}}-\frac{1}{\mu_\mathrm{D}}\right)\lambda}\right] \geq 0,\end{aligned} \tag{8}$$

and

$$\frac{\partial^2\bar{A}}{\partial W^2} = \frac{1}{2}\lambda e^{-\left(W-\frac{1}{\mu_\mathrm{U}}-\frac{1}{\mu_\mathrm{D}}\right)\lambda} \geq 0, \tag{9}$$

which proves the monotone convexity. Furthermore, as the update window size increases (i.e., $W\to\infty$), the first-order derivative becomes 1/2 and the second-order derivative goes to zero, indicating the asymptotic linear relationship.

Furthermore, the result of (7) also reveals that the proposed algorithm can regulate the AoI under different traffic loads. By taking the first- and second-order derivatives of $\bar{A}$ with respect to λ, we have

$$\begin{aligned}\frac{\partial\bar{A}}{\partial\lambda} &= \frac{\partial\bar{A}}{\partial\bar{N}}\frac{\partial\bar{N}}{\partial\lambda} = -\frac{1}{2}\left[-\frac{1}{\bar{N}^2}\left(1-e^{-\bar{N}}\right)+\frac{1}{\bar{N}}e^{-\bar{N}}\right] \\ &= -\frac{e^{-\bar{N}}}{2\bar{N}^2}\left[1+\bar{N}-e^{\bar{N}}\right] \geq 0,\end{aligned} \tag{10}$$

and

$$\begin{aligned}\frac{\partial^2\bar{A}}{\partial\lambda^2} &= -\left(\frac{e^{-\bar{N}}}{\bar{N}^3}+\frac{e^{-\bar{N}}}{2\bar{N}^2}\right)\left(e^{\bar{N}}-\bar{N}-1\right)+\frac{e^{-\bar{N}}}{2\bar{N}^2}\left(e^{\bar{N}}-1\right) \\ &= -\frac{e^{-\bar{N}}}{\bar{N}^3}\left(e^{\bar{N}}-1-\bar{N}-\frac{\bar{N}^2}{2}\right) \leq 0,\end{aligned} \tag{11}$$

where $\bar{N} = \lambda\left(W-\frac{1}{\mu_\mathrm{U}}-\frac{1}{\mu_\mathrm{D}}\right)$. Thus, the average AoI $\bar{A}$ increases with the traffic intensity in a convex manner. In specific, the first- and second-order derivatives both go to zero as $\bar{N}\to\infty$, indicating that the AoI finally levels off as the traffic load further increases.

Notice that

$$\lim_{\lambda\to\infty}\bar{A} = \frac{1}{2}\left(W+\frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}\right), \tag{12}$$

and

$$\begin{aligned}\lim_{\lambda\to 0}\bar{A} &= \frac{1}{2}\left(W+\frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}\right)-\frac{1}{2}\left(W-\frac{1}{\mu_\mathrm{U}}-\frac{1}{\mu_\mathrm{D}}\right) \\ &= \frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}},\end{aligned} \tag{13}$$

according to (7). Thus, the range of average AoI is constrained within $\left[\frac{1}{2}\left(\frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}\right), \frac{1}{2}\left(W+\frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}\right)\right]$ regardless of the traffic load. Therefore, the average AoI is regulated for the given update window size.

D. Service Delay Analysis

According to the M/M/1 queueing model, the average service delay is given by

$$\bar{D} = \frac{1}{\frac{1}{\mathbb{E}[X]}-\lambda}. \tag{14}$$

Substitute Eq. (2) into (14), and the average service delay can be obtained:

$$\bar{D} = \frac{\frac{p}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}}{1-\frac{p\lambda}{\mu_\mathrm{U}}-\frac{\lambda}{\mu_\mathrm{D}}}, \tag{15}$$

where p is given by Eq. (4).

The result of (15) shows that the average delay increases with the update probability. As the update probability decreases with update window size W, increasing update window size helps to reduce the average delay. However, the average AoI increases with update window size according to (7). Therefore, there exists a tradeoff relationship between the average AoI and service delay with respect to the update window size.

In case of strict AoI requirements posed by fast time-varying contents, the update window size should be set to the minimal value and the average delay achieves the maximum:

$$\lim_{W\to\frac{1}{\mu_\mathrm{U}}+\frac{1}{\mu_\mathrm{D}}}\bar{D} = \lim_{p\to 1}\bar{D} = \frac{1}{\frac{\mu_\mathrm{U}\mu_\mathrm{D}}{\mu_\mathrm{U}+\mu_\mathrm{D}}-\lambda} \triangleq \bar{D}_{\max}. \tag{16}$$

In this case, all requests trigger cache update, bringing heavy traffic load to the BS. The other extreme case is the static contents which have no AoI requirement. The update window size can be set to infinity, and the corresponding average delay is given by

$$\lim_{W\to\infty}\bar{D} = \lim_{p\to\infty}\bar{D} = \frac{1}{\mu_\mathrm{D}-\lambda} \triangleq \bar{D}_{\min}. \tag{17}$$

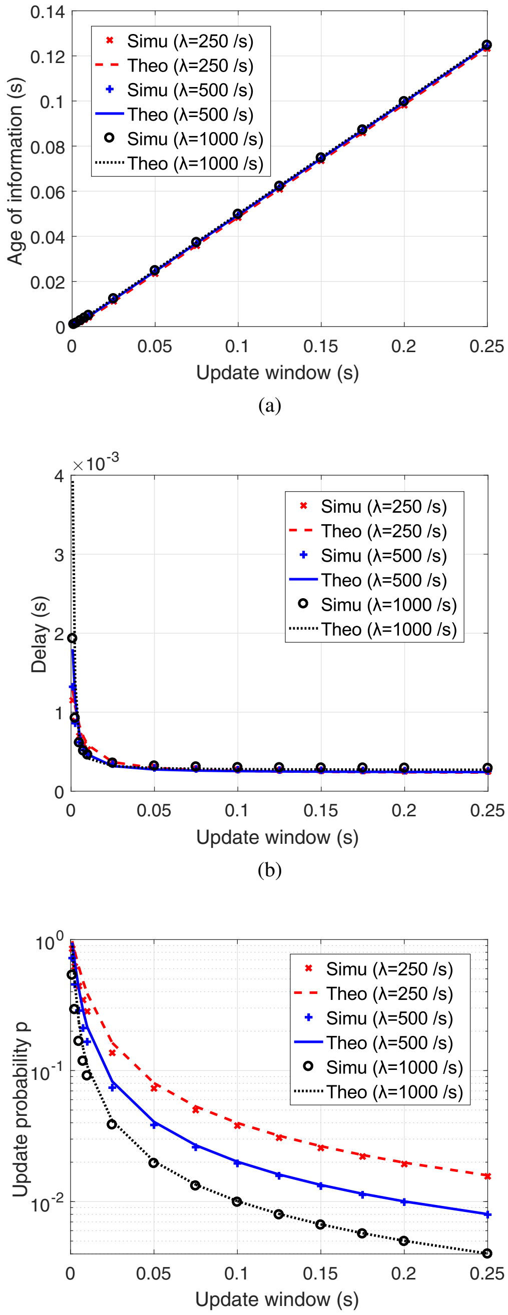
In this case, the cache is merely updated and the delay achieves the minimum. Eqs. (16) and (17) indicate how much delay can be traded by sacrificing the content freshness. In general, the gain of AoI-delay trading is more significant in heavily-loaded networks with lower transmission rate of source nodes. In addition, the delay reduction depends on the specific AoI requirement of the content type. Furthermore, the AoI-delay trading suffers from the marginal effect. In specific, the average AoI increases with the update window size linearly while the average delay levels off at the minimum, when the update window size becomes sufficiently large ($W \to \infty$).

V. Simulations

This section conducts system-level simulations to validate the analytical results by implementing the OMNeT++ simulator. The derived update probability, average AoI and service delay are compared with the simulation results. The tradeoff relationship between AoI and delay is revealed with respect the update window size. In addition, the influence of traffic load on the AoI and delay performance is also illustrated.

Monte Carlo method is applied in the OMNeT++ simulation, where the user locations, content requests, and transmission time are randomly generated. The BS implements the proposed freshness-aware cache update scheme upon the content requests. The average service rate of the BS is set to μ_{U} = 1000 contents/s for the uplink cache update, and μ_{D} = 5000 contents/s for downlink content delivery. Continuous-time simulations are conducted under different update window sizes and traffic loads, whereby the average AoI, delay and updated probability are calculated throughout the simulations. The simulation results and analytical ones are compared as shown in Fig. 3 which reveals three facts. Firstly, the analytical results are shown to be close to the simulation ones, validating the approximated analysis of AoI, service delay and update probability. Secondly, both the analytical and simulation results clearly show that the average delay decreases while the AoI increases with the update window size, revealing the tradeoff relationship. In specific, the AoI increases almost linearly with the update window size, which is consistent with the analysis. The delay firstly decreases but finally levels off. The reason is that the probability that a content request triggers cache update decreases as the update window size increases, as show in Fig. 3(c). Accordingly, the total wireless transmission load of the BS decreases, improving the delay performance. When the update window size is large enough, the cache is merely updated and the total transmission load achieves minimal. In this case, increasing update window size cannot further decrease delay, whereas the AoI keeps increasing linearly. Thirdly, the average AoI and delay differ little under different traffic loads, while the update probability is show to decrease with the traffic load in Fig. 3(c). These results are consistent with the theoretical analysis, i.e., the proposed freshness-aware content update scheme can restrain frequent cache update in case of high content request rates.

Figure 4 shows the influence of traffic load on AoI and delay performance. The average AoI increases with traffic load and converges to a constant when the traffic load is large, as shown in Fig. 4(a). When the traffic load is extremely low, the cache is updated upon user requests with high probability, and the AoI is close to the two-hop transmission time. In this case, the update cycle length is random and mainly relies on the arrival of content requests, whereas the update window size has little influence. When the traffic load is sufficiently high, the cache will be regularly updated with cycle length equal to the update window size, since there are always requests to serve. In this case, the AoI mainly depends on the update window size instead of the traffic load. The average delay



(a)

(b)

(c)

Fig. 3: Analytical results evaluation with respect to update window size, (1) age of information, (2) service delay, and (3) update probability.

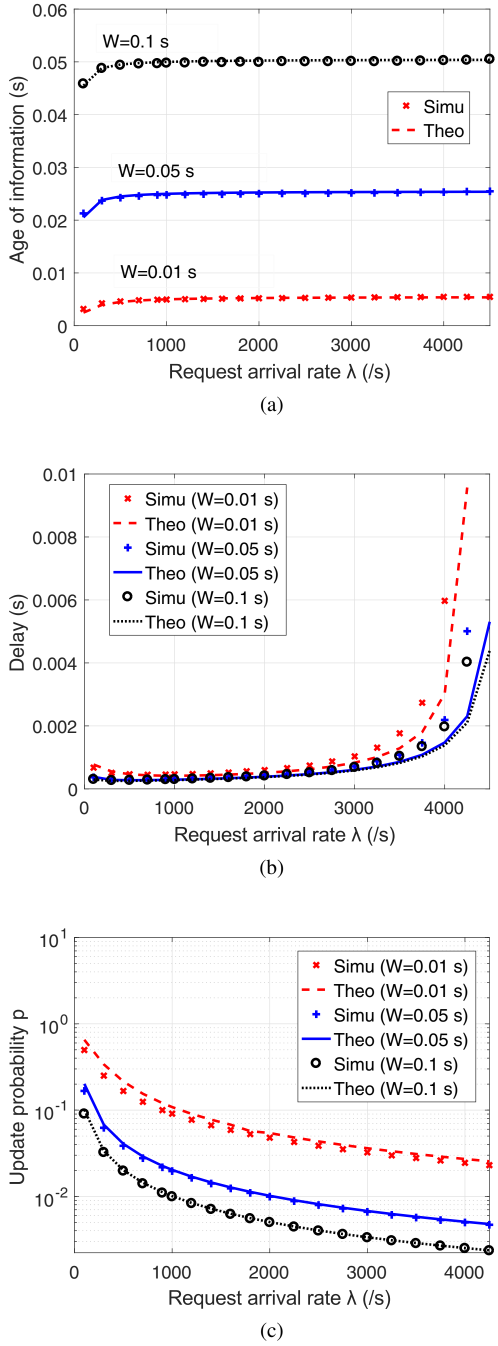


Fig. 4: Influence of traffic load, (1) age of information, (2) service delay, and (3) update probability.

is shown to increase with the traffic load in Fig. 4(b). This is because that the total transmissions still increases with the traffic load, although the update probability decreases as shown in Fig. 4(c).

The important insights of Figs. 3 and 4 are three-fold: (1) The delay performance can be enhanced by sacrificing the freshness of contents, especially in heavily loaded networks; (2) The proposed freshness-aware content update scheme can restrain frequent update in case of high request arrival rates, by setting the update window size appropriately; (3) The delay-AoI tradeoff suffers from the marginal effect, and the update window size should be carefully devised according to the content type and QoS requirements.

VI. Conclusions and Future Work

This work has investigated the fundamental tradeoff relationship between AoI and service delay in the mobile edge caching networks, where the constrained transmission resources are used for both cache update and content delivery at the BS. A freshness-aware cache update scheme has been devised, whereby the BS updates the cached contents based on the AoI upon user requests. The approximated average AoI and delay have been derived in closed forms under the proposed scheme, and validated by the OMNeT++ simulations. Both the analytical and simulation results have demonstrated the tradeoff relationship between AoI and delay, which can be flexibly adjusted by tuning the update window size under the proposed scheme. In addition, the proposed scheme is shown to regulate the AoI and delay performance under different traffic loads, by restraining frequent cache update in case of high request rates. The proposed cache update scheme can be utilized in practical mobile caching systems providing time-varying content services, whereby users can obtain the fresh and effective information within short delay. For the future work, the update window size will be optimized, considering the coexistence of multiple-type contents and differentiated AoI-delay requirements.

References

[1] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5G wireless networks,” *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[2] E. Baştuğ, M. Kountouris, M. Bennis, and M. Debbah, “On the delay of geographical caching methods in two-tiered heterogeneous networks,” in *IEEE SPAWC’16*, Edinburgh, UK, Aug. 2016, pp. 1–6.

[3] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, and X. Shen, “Cooperative edge caching in user-centric clustered mobile networks,” *IEEE Trans. Mobile Comput.*, vol. 17, pp. 1791–1805, Aug. 2018.

[4] Q. Kuang, J. Gong, X. Chen, and X. Ma, “Age-of-information for computation-intensive messages in mobile edge computing,” *arXiv preprint arXiv:1901.01854*, 2019.

[5] R. D. Yates, M. Tavan, Y. Hu, and D. Raychaudhuri, “Timely cloud gaming,” in *IEEE INFOCOM’17*, Atlanta, GA, USA, May 2017, pp. 1–9.

[6] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, “Information freshness and popularity in mobile caching,” in *IEEE ISIT’17*, Aachen, Germany, Jun. 2017, pp. 136–140.

[7] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, “Minimizing age of information in vehicular networks,” in *IEEE SECON’11*, Salt Lake City, UT, USA, Jun. 2011, pp. 350–358.

[8] H. Hsu and K. Chen, “Optimal caching time for epidemic content dissemination in mobile social networks,” in *IEEE ICC’16*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.

[9] H. Ahlehagh and S. Dey, “Video-aware scheduling and caching in the radio access network,” *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.

[10] S. Li, J. Xu, M. van der Schaar, and W. Li, “Trend-aware video caching through online learning,” *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2503–2516, Dec. 2016.

[11] R. D. Yates, P. Ciblat, A. Yener, and M. Wigger, “Age optimal constrained cache updating,” in *IEEE ISIT’17*, Aachen, Germany, Jun. 2017, pp. 141–145.