Proactive Caching for Low Access-Delay Services under Uncertain Predictions

Ran Liu

Electrical and Computer Engineering
Northeastern University
Boston, MA, USA
rliu1@ece.neu.edu

Edmund Yeh

Electrical and Computer Engineering
Northeastern University
Boston, MA, USA
eyeh@ece.neu.edu

Atilla Eryilmaz

Electrical and Computer Engineering
Ohio State University
Columbus, OH, USA
eryilmaz.2@osu.edu

ABSTRACT

Network traffic for delay-sensitive services has become a dominant part in the network. Proactive caching with the aid of predictive information has been proposed as a promising method to enhance delay performance. In this paper, we analytically investigate the problem of how to efficiently utilize uncertain predictive information to design proactive caching strategies with provably good access-delay characteristics. We first derive an upper bound for the average amount of proactive service per request that the system can support. We then analyze the behavior of a family of threshold-based proactive strategies using an innovative Markov chain analysis, and show that the average amount of proactive service per request can be maximized by properly selecting the threshold. Finally, we propose the UNIFORM strategy, which is the threshold-based strategy with the optimal threshold. Surprisingly, we show that it outperforms the commonly used Earliest-Deadline-First (EDF) type proactive strategies in terms of delay. We perform extensive numerical experiments to demonstrate the influence of thresholds on delay performance, and explicitly compare performance of the EDF strategy and the UNIFORM strategy.

KEYWORDS

Proactive Caching; Prefetching; Queueing Theory; Markov Chain

ACM Reference Format:

Ran Liu, Edmund Yeh, and Atilla Eryilmaz. 2019. Proactive Caching for Low Access-Delay Services under Uncertain Predictions. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '19 Abstracts)*, June 24–28, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3309697.3331471>

1 BACKGROUND AND MOTIVATION

Proactive caching techniques are seen as an effective method to improve the delay performance of delay-sensitive services in communication networks. Specifically, proactive caching techniques take advantage of predictive information of user requests and network states, utilize spare bandwidth resources and potentially place data objects in caches before requests are generated.

Our work aims to study the characteristics of proactive caching based on uncertain predictive information from a fundamental

queueing theory perspective. In contrast to [2–4], we not only examine the basic queueing dynamics of the proactive system but also explore how to strategically utilize uncertain predictions to enhance delay performance. In terms of delay performance, we will show improvements relative to the Earliest-Deadline-First (EDF) type strategy, which has been widely used in network scheduling problems.

2 NETWORK MODEL

We consider a single-queue system with one server providing delay-sensitive services to the user. The system operates in continuous time from time 0. The user receives service from the server at a constant rate of μ bits/sec.

Request Processes: Requests arrive at the server, requesting same-sized data objects of s bits. The *Potential Request Process* is a Poisson Process $\{P(t); t > 0\}$ with an overall arrival rate of λ , where the i th arrival, i.e. *Potential Request* i , requests object $r_i \in \mathbb{Z}^+ = \{1, 2, 3, \dots\}$ at time $t_i \in \mathbb{R}^+$, where $0 < t_1 < t_2 < \dots$. The *Actual Request Process* $\{A(t); t > 0\}$ is a thinned version of $P(t)$ where each arrival on $P(t)$ is an arrival on $A(t)$ with probability p , independent of all other arrivals. Let $\{R_i; i = 1, 2, \dots\}$ be IID Bernoulli(p) indicator random variables where $R_i = 1$ if the i th arrival on $P(t)$ is an arrival on $A(t)$. We assume that every potential request requests a different object, i.e. $r_i \neq r_j, \forall i \neq j$, i.e., the catalog size is assumed to be infinite.

Predictions: At time 0, the server is assumed to know the sequence of objects $(r_1, r_2, r_3, \dots)$ to be requested by the arrivals in $\{P(t)\}$, and the probability p . It has no prior knowledge of the precise arrival epochs $\{t_i\}$, or the realizations of the indicator random variables $\{R_i\}$. The server observes $A(t)$ but not $P(t)$. At time $t > 0$, the sequence of indices for future potential requests from the server's viewpoint, or the *prediction window*¹, is:

$$\Pi(t) = (I(t) + 1, I(t) + 2, I(t) + 3, \dots) \quad (1)$$

where $I(t)$ is defined as:

$$I(t) \triangleq \max \{i | t_i < t, R_i = 1\} \quad (2)$$

i.e. the index of the most recent actual request before time t . The server proactively works on request i only if $i \in \Pi(t)$ at time t . More details regarding the assumptions on the network model and predictions model can be found in [1].

Proactive Strategies: In a *reactive scheme*, the server works only on requests made by actual request arrivals. We focus on

¹We assume the prediction window size to be infinite for simplicity of analysis. This assumption guarantees that the server always has predicted requests on which to do proactive work.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMETRICS '19 Abstracts, June 24–28, 2019, Phoenix, AZ, USA

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6678-6/19/06.

<https://doi.org/10.1145/3309697.3331471>

proactive schemes, where the server works on future potential requests when not serving requests made by actual requests. The server can proactively send a data object, partially or in entirety, to the user, which can store the data object in a local cache.²

Let $U_i(t) \leq s$ be the proactive work done for request i by time t , i.e. the number of bits of object r_i sent to the user and stored in the cache by time t . Let $U_i \triangleq \min\{s, U_i(t_{H(i)})\}$ be the total proactive work done for request i , where $H(i) \triangleq \min\{j \geq i | R_j = 1\}$ represents the first potential request after i which is realized. Let $\mathbf{U}(t) \triangleq (U_{I(t)+1}(t), U_{I(t)+2}(t), U_{I(t)+3}(t), \dots)$ be the set of $U_i(t)$'s where $i \in \Pi(t)$. $V(t)$ is defined as the total number of bits for actual requests waiting to be transmitted in the queue at time t . At time t , based on $\mathbf{U}(t)$, the prediction window $\Pi(t)$ and the queue size $V(t)$, a stationary proactive rate allocation strategy Ψ_P at the server is defined as:

$$\Psi(V(t), \Pi(t), \mathbf{U}(t)) = \{\rho_V(t), \rho_{I(t)+1}(t), \rho_{I(t)+2}(t), \rho_{I(t)+3}(t), \dots\}$$

where $\rho_V(t)$ is the rate allocated to serve the queue of $V(t)$, and $\rho_{I(t)+i}(t)$, $i \geq 1$, is the rate allocated to fetch object $r_{I(t)+i}$ at time t . We consider the set Γ_P of proactive strategies Ψ_P satisfying:

- (Reactive State) If $V(t) > 0$: $\rho_V(t) = \mu$, $\sum_{i=1}^{\infty} \rho_{I(t)+i} = 0$
- (Proactive State) If $V(t) = 0$: $\rho_V(t) = 0$, $\sum_{i=1}^{\infty} \rho_{I(t)+i} = \mu$
- The limiting average amount of proactive work received per potential request and per actual request, respectively:

$$\bar{U} \triangleq \lim_{t \rightarrow \infty} \frac{\sum_{i=1}^{I(t)} U_i}{I(t)}, \quad \bar{U}_A \triangleq \lim_{t \rightarrow \infty} \frac{\sum_{i \in \mathbb{Z}^+: i \leq I(t), R_i=1} U_i}{A(t)} \quad (3)$$

exist for Ψ_P ;

Problem Formulation: Given λ , μ , p and s , our optimization problem then can be formulated as:

$$\underset{\Psi_P \in \Gamma_P}{\text{maximize}} \quad \bar{U}(\Psi_P) \quad (4)$$

Let Ψ^* be an optimal solution to problem (4) and let $\bar{U}_{max} \triangleq \bar{U}(\Psi^*)$ denote the $\bar{U}$ achieved by Ψ^* .

We denote the average delay per actual request under Ψ^* by $\bar{D}_{\Psi^*}$. We will derive the closed-form expression of $\bar{D}_{\Psi^*}$, and analytically demonstrate its improvement relative to average delay of the EDF proactive strategy.

3 MAIN RESULT

We first define two important properties of proactive strategies:

DEFINITION 1. A proactive strategy $\Psi_P \in \Gamma_P$ satisfies Property 1 if the following condition is satisfied:

$$\lim_{t \rightarrow \infty} \frac{\sum_{i=I(t)+1}^{\infty} U_i(t)}{t} = 0, \quad w.p. 1 \quad (5)$$

DEFINITION 2. A proactive strategy satisfies Property 2 if the following condition is satisfied:

$$\bar{U}_A = \bar{U}, \quad w.p. 1 \quad (6)$$

We now derive an upper bound of the limiting average proactive work that can be done by a proactive strategy $\Psi_P \in \Gamma_P$ as:

COROLLARY 1. Given μ , λ , s and p satisfying $\frac{\mu}{s} \leq \lambda < \frac{\mu}{ps}$, the limiting average amount of proactive work per potential request under strategy $\Psi_P \in \Gamma_P$ satisfies

$$\bar{U} \leq \frac{\mu - p\lambda s}{\lambda(1-p)} \triangleq U^*, \quad w.p. 1 \quad (7)$$

where equality holds in (7) if and only if strategy Ψ_P satisfies both Property 1 and Property 2.

In order to construct an optimal proactive strategy to solve (4), we first define a family of threshold-based strategies Ψ_P^ϕ in Γ_P . The detailed definition of Ψ_P^ϕ can be found in [1]. Summarily speaking, Ψ_P^ϕ selects a threshold $\phi \in (0, s]$, and proactively works on request $J(t) = \min\{i \in \Pi(t) | U_i(t) < \phi\}$ when in proactive state at t .

In order to analyze the proactive system, we construct a Markov chain by considering the system state at specific epochs and carry out the corresponding analysis, the details of which can be found in [1]. We prove that the threshold-based strategy Ψ_P^ϕ gives an optimal solution to problem (4) if and only if $\phi = U^*$, as in the following Corollary:

COROLLARY 2. The objective in (4) is maximized with a threshold-based proactive strategy Ψ_P^ϕ if and only if $\phi = U^*$.

We call $\Psi_P^{U^*}$ the UNIFORM strategy. We then derive the closed-form expression for limiting average delay per request under the UNIFORM strategy. Next, this is compared to the limiting average delay per request under the EDF strategy, as in the following Corollary:

COROLLARY 3. Given μ , λ , s and p satisfying $\frac{\mu}{s} \leq \lambda < \frac{\mu}{ps}$, the average delay $\bar{D}_U$ of the UNIFORM strategy is no greater than the average delay $\bar{D}_E$ of the EDF strategy with probability 1:

$$\bar{D}_U \leq \bar{D}_E, \quad w.p. 1 \quad (8)$$

The equality holds if and only if $p = 0$.

We conduct extensive simulations to show the delay performance of the threshold-based strategies with different thresholds. The results show that proactive caching not only greatly improves delay performance in lightly-loaded cases as found in [4], but also, under the UNIFORM strategy, works exceedingly well in heavily-loaded scenarios.

REFERENCES

- [1] Ran Liu, Edmund Yeh, and Atilla Eryilmaz. 2019. Proactive Caching for Low Access-Delay Services Under Uncertain Predictions. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 1, Article 2 (March 2019), 46 pages. <https://doi.org/10.1145/3322205.3311073>
- [2] Leela Srikanth Muppirisetty, John Tadrous, Atilla Eryilmaz, and Henk Wymeersch. 2015. On proactive caching with demand and channel uncertainties. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 1174–1181. <https://doi.org/10.1109/ALLERTON.2015.7447141>
- [3] John Tadrous and Atilla Eryilmaz. 2016. On Optimal Proactive Caching for Mobile Networks With Demand Uncertainties. *IEEE/ACM Transactions on Networking* 24, 5 (October 2016), 2715–2727. <https://doi.org/10.1109/TNET.2015.2478476>
- [4] Shaoquan Zhang, Longbo Huang, Minghua Chen, and Xin Liu. 2017. Proactive Serving Decreases User Delay Exponentially: The Light-Tailed Service Time Case. *IEEE/ACM Trans. Netw.* 25, 2 (April 2017), 708–723. <https://doi.org/10.1109/TNET.2016.2607840>

² Since our focus is on the effects of uncertain predictions, we assume for simplicity that the cache size is infinite.