# Age of Information aware caching updating

Philippe Ciblat

*Joint work with Haoyue Tang, Michèle Wigger, and Roy Yates*

Fundings: ERC Starting Grant "CTOCom" and NSF award "CIF-1422988"

# Age of Information

We consider time-sensitive file :

- the content of the file depends on the time
- Ex : newspaper website, web crawling, video last version, ...

**Metric : Age of Information**

- Freshness of the information of a file is captured by the **Age of Information** (AoI),
- defined as the time elapsed since its last modification
- $X_n(t)$ age of content/file $n$ at timeslot $t$

**Remark :** AoI different from the delay since the transmitted file is not fresh (ie, not the last update) even if delay transmission is zero.

# File popularity

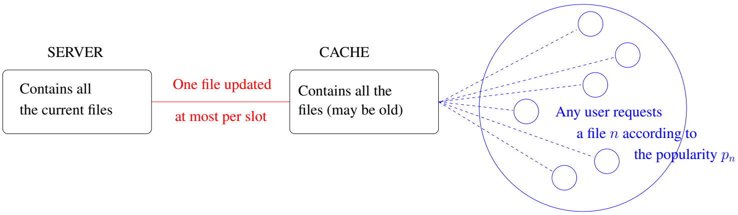
- Each content has its own popularity (probability to be requested)
- $p_n(t)$ popularity of content/file $n$ at timeslot $t$

**Example :** Zipf distribution of parameter $s \Rightarrow$ normalized request frequency of file ranked $k$ out of $N$

$$f(k) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s}$$

- $s \to 0$ : uniform distribution
- $s \to \infty$ : only one file is requested

# Problem statement




- When a user requests an item, the cache sends its local version.
- Issue : this version can be outdated since each item is time-varying, and the capacity-constrained server-cache link does not enable us to provide the latest version.

## Question

- How should items be downloaded from the server in order to be as up-to-date as possible ?
- Equivalently, how should the server push updated versions to the cache s.t. users receive the most recent versions they request ?

# Applications

- Data-Base Context
  - Content items = records in a database (server),
  - Cache is the local copy.

- Cloud Radio Access Network (RAN)
  - Server = BBU in Cloud RAN or standard BS.
  - Cache = small-cell base station (RRH in Cloud-RAN) delivering popular content to nearby mobile users.
  - Server-Cache link is rate-limited whereas cache-user links are short-range and ultra-high data rate.

- Satellite based Broadcasting System
  - Server = satellite broadcasting the same content updates to thousands of caches.
  - Cache = local storage in a TV/video news distribution

- Web crawling
  - Indexing engine for a web portion

# Mathematical model

- File $n$ is requested from the cache with probability $p_n > 0$, (here time-invariant).

$$\mathbf{p} = [p_1, \cdots, p_N] \quad \text{(popularity vector)}$$

- Cache is able to download $K$ files from the server within $T$ slots

$$\lambda = \lim_{T \to \infty} \frac{K}{T} \quad \text{(update rate)}$$

- In slot $t$, at most *single* file $u_t$ may be updated from the server ($u_t$= 0 if no updated file). Let $T$ be the number of slots.

$$\mathbf{u} = [u_1, \cdots, u_T] \quad \text{(update vector)}$$

# Optimization problem

The average age of a randomly requested item from the cache is

$$\overline{X}(\mathbf{u}) = \sum_{n=1}^{N} p_n \overline{X}_n(\mathbf{u}) \text{ with } \overline{X}_n(\mathbf{u}) = \frac{1}{T}\int_0^T X_n(t)\, dt$$

since a request for item $n$ is uniformly-distributed over $[0, T]$.

**Optimization problem : update scheduling**

$$\overline{X}^*(K, T) = \min_{\mathbf{u}} \overline{X}(\mathbf{u})$$

s.t.

- $u_t \in \{1, \cdots, N\}$ for all $t$,
- $\sum_{t=1}^{T} \mathbf{1}\{u_t > 0\} = K$.

# Optimization problem

The average age of a randomly requested item from the cache is

$$\overline{X}(\mathbf{u}) = \sum_{n=1}^{N} p_n \overline{X}_n(\mathbf{u}) \text{ with } \overline{X}_n(\mathbf{u}) = \frac{1}{T} \int_0^T X_n(t)\, dt$$

since a request for item $n$ is uniformly-distributed over $[0, T]$.

**Optimization problem : update scheduling**

$$\overline{X}^*(K, T) = \min_{\mathbf{u}} \overline{X}(\mathbf{u})$$

s.t.

- $u_t \in \{1, \cdots, N\}$ for all $t$,
- $\sum_{t=1}^{T} \mathbf{1}\{u_t > 0\} = K$.

**Intractable combinatorial optimization problem**

# Problem simplification : per-file update rate

- Let $k_n$ be the number of update for file $n$ with equal inter-update time.
- Consider $\lambda_n = \lim_{T\to\infty} k_n/T$ the per-file update rate.
- Actually, we get

$$\overline{X}_n \approx \frac{1}{\lambda_n}$$

## New optimization problem

$$\min_{\lambda_1,\dots,\lambda_N} \sum_{n=1}^{N} \frac{p_n}{\lambda_n}$$

s.t.

- $\lambda_n \geq 0$,
- $\lambda_1 + \cdots + \lambda_N = \lambda$.

# Final results

## Main result 1

Problem is convex and leads to

$$\lambda_n^* = \frac{\lambda\sqrt{p_n}}{\sum_{i=1}^N \sqrt{p_i}}$$

**Update rate of file $n$ follows a square-root law wrt. its popularity**

## Main result 2

The minimum average age is

$$\overline{X}^* = \frac{\Delta^*(\mathbf{p})}{\lambda} + 1,$$

with

$$\Delta^*(\mathbf{p}) = \frac{1}{2}\left(\sum_{i=1}^N \sqrt{p_i}\right)^2.$$

# Practical protocol

Let $\overline{\tau}_n^\star = 1/\lambda_n^\star$ be the optimal inter-update time for file $n$

$$n_0(t) = \arg\max_{n\in\{1,\cdots,N\}} \underbrace{(X_n(t) - \overline{\tau}_n^\star)}_{\text{Schedule-ordered by Age-based Priority (SOAP)}}$$

**General context :**

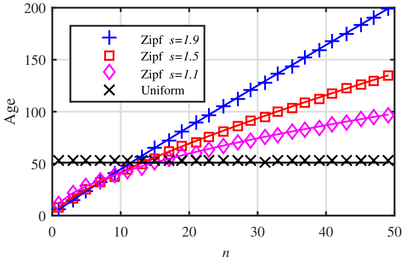
- $r(D, X)$ : rank function with descriptor $D$ and age $X$
- Schedule user set $\mathcal{N}_0$ (with $|\mathcal{N}_0| < K$)

$$\mathcal{N}_0(t) = \arg\max_{n\in\{1,\cdots,N\}} r(D_n(t), X_n(t))$$

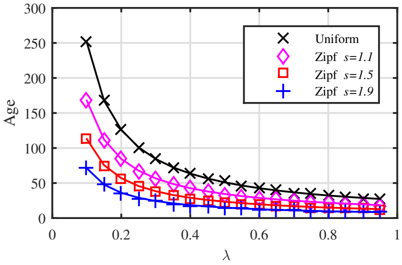
- Many policies follow this shape
  - Round-Robin (RR), $r(\emptyset, X_n) = X_n$
  - weighted Round-Robin, $r(d_n, X_n) = d_n.X_n$

# Some numerical results

- $N = 50$ files, $\lambda = 0.5$




Age for each file $n$




Average Age vs. $\lambda$

- Proposed policy reduces the average age of more popular items at the expense of less popular items
- When Zipf parameter $s$ increases, update rates optimization exploits the concentration in the popularities better.

# Extension 1

**New :**

- The amount of data to download (from the server to the cache) depends on the file <u>and</u> the **age** of the file
- Let $f_n(X)$ be the time spent to update the file $n$ with age $X$. Assumed to be strictly positive, bounded, non-decreasing, concave, and differentiable over $\mathbb{R}_+$.

**Main assumption :**

- Uniform inter-update duration $\overline{\tau}_n$
- Let

$$\lambda_n := \frac{f_n(\overline{\tau}_n)}{\overline{\tau}_n}$$

  be the *file utilization ratios*, i.e., the fraction of time that each file is being updated.

# Extension 1 : problem resolution

- $g_n(t) := f_n(t)/t$.
- $g_n(.)$ strictly decreasing and its image is $(0, \infty)$.
- $g_n(.)$ has an inverse function denoted by $g_n^{(-1)}(.)$, which is also strictly decreasing, and $\overline{\tau}_n = g_n^{(-1)}(\lambda_n)$.

$$\min_{\{\lambda_n\}_n} \sum_{n=1}^{N} p_n \cdot \underbrace{g_n^{-1}(\lambda_n)\left(\frac{1}{2} + \lambda_n\right)}_{h_n(\lambda_n)}$$

s.t. $\lambda_n \geq 0,\ \forall n$, and $\lambda_1 + \cdots + \lambda_N \leq 1$.

**Result**

- $h_n(\cdot)$ is strictly decreasing.
- Monotonic optimization framework [Jorswieck2010]
- Optimal solution can be found using the so-called Branch-Reduce-Bound (BRB)

# An example for $f_n(\cdot)$

- In each timeslot, a certain portion of each file becomes obsolete, and the cache and the server know the obsolete parts.
- These bits can thus be modeled as erasures (as should be replaced with new unknown ones).

Assuming Binary Erasure Channel (BEC) with parameter $\Delta_n$, and a file of size $B_n$, the average number of erased positions in the file after $t$ timeslots is $B_n(1-(1-\Delta_n)^t)$. To avoid $\lim_{t\to 0} f_n(t) = 0$, we force

$$f_n(t) := B_n - (B_n - \varepsilon_n)(1-\Delta_n)^t.$$

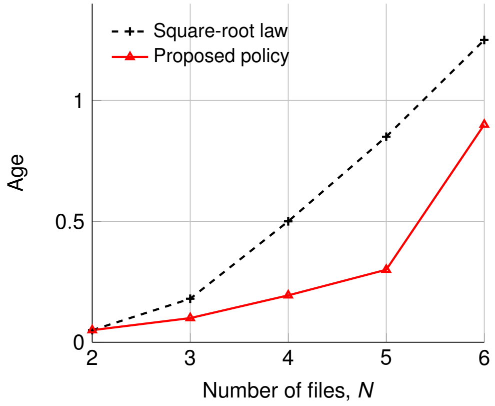
**Here, convex optimization problem (KKT)**

$$\lambda_n^\star = h_n'^{(-1)}\left(-\frac{\nu^\star}{p_n}\right), \quad \forall n.$$

with the “waterlevel" $\nu^\star \geq 0$ chosen such that $\sum_{n=1}^{N} \lambda_n^\star = 1$.

# Some numerical results

- $f_n(.)$ obtained with BEC model with $B_n$ and $\varepsilon_n = 0.02$
- Popularity follows a Zipf-distribution with parameter $\alpha = 1.8$.



# Extension 2

**New :**

- Popularity is time-varying : $p_n(t)$
- $p_n(t) \in \{R_1, \cdots, R_N\}$, and Markov chain for modelling the variation

$$\mathrm{Prob}(p_n(t+1) = R_m | p_n(t) = R_\ell) = q_{\ell,m}$$

and the cost is $\omega(R_\ell)$ if $p_n(t) = R_\ell$.

**Problem (to be solved) :** find $\pi(\{X_n\}_n, \{p_n\}_n)$ be the updating policy

$$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}\in\Pi} \lim_{T\to\infty} \mathbb{E}_{\boldsymbol{\pi}} \left[ \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \omega_n(R_{n,t}) X_{n,t} \right],$$

s.t. $\sum_{n=1}^{N} u_{n,t} \leq M, \quad \forall t.$

(Finite-State) Markov Decision Process (MDP) : but **not scalable**

# Extension 2 : problem resolution

Relaxing hard constraint on the update number.

$$\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}\in\Pi} \lim_{T\to\infty} \mathbb{E}_{\boldsymbol{\pi}} \left[ \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \omega_n(R_{n,t}) X_{n,t} \right],$$

s.t. $\lim_{T\to\infty} \mathbb{E}_{\boldsymbol{\pi}} \left[ \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} u_{n,t} \right] \leq M$.

Constrained (Finite-State) Markov Decision Process :

- still **not scalable** but partially **factorizable**,
- and replace deterministic policy with a **stationary policy**.

Using Lagrangian function :

$$\begin{aligned}\mathcal{L}(\pi, W) &= \lim_{T\to\infty} \mathbb{E}_{\pi}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N}(\omega_n(R_{n,t})X_{n,t} + W(u_{n,t} - M))\right] \\ &= \sum_{n=1}^{N} \underbrace{\lim_{T\to\infty} \mathbb{E}_{\pi}\left[\frac{1}{T}\sum_{t=1}^{T}\omega_n(R_{n,t})X_{n,t} + W(u_{n,t} - M)\right]}_{\mathcal{L}_n(\pi_n, W)}\end{aligned}$$

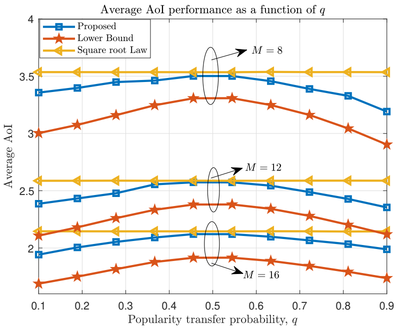
- Given $W$, solve each $\mathcal{L}_n$ for obtaining $\pi_n$
  (Finite-state) MDP with scalable number of states (since working file by file)
- Find optimal $W$ with an exhaustive 1-D search (not necessary unique) [Beutler1985]

# Some numerical results

Two popularity modes $\mathcal{R} = \{1, 2\}$ such that all files $n$ have following transition matrix

$$\mathbf{T} = \begin{bmatrix} q & 1-q \\ 1-q & q \end{bmatrix}, \quad \forall n \in \{1, \ldots, N\}.$$

with $\omega_n(1) = 0.2\overline{\omega}_n$ and $\omega_n(2) = 1.8\overline{\omega}_n$, where $\overline{\omega}_n \propto 1/n^s$ with $s = 1.5$.




Age vs $q$ for different $M$ and $N = 64$.

# Perspectives

Main open problem : finding a simple SOAP for the last extension

- Idea : using the heuristic of the Whittle's index (WI).
  Finding $W$ in Bellman's equation with cost $\mathcal{L}_n$ such that both actions (to schedule/not to schedule) are equivalent
- If popularity is not time-varying

$$\text{WI} : n_0(t) = \arg \max_{n \in \{1, \cdots, N\}} \sqrt{p_n} X_n(t)$$

  Weighted RR with $d_n = \sqrt{p_n}$. Close to square-root law ?
- Difficult for extension 2 since 2-D state

Related Publications :


- H. Tang, P. Ciblat, J. Wang, M. Wigger, and R. Yates : Cache updating strategy minimizing Age of Information with time-varying files' popularity, IEEE ITW, 2021.
- H. Tang, P. Ciblat, J. Wang, M. Wigger, and R. Yates : Age of Information aware cache updating with file- and age-dependent update durations, Wiopt, 2020.
- R. Yates, P. Ciblat, A. Yener, and M. Wigger : Age-optimal constrained cache updating , IEEE ISIT, 2017.