Optimal placement for coded caching at the edge

Valerio Bioglio, Frederic Gabry, and Ingmar Land

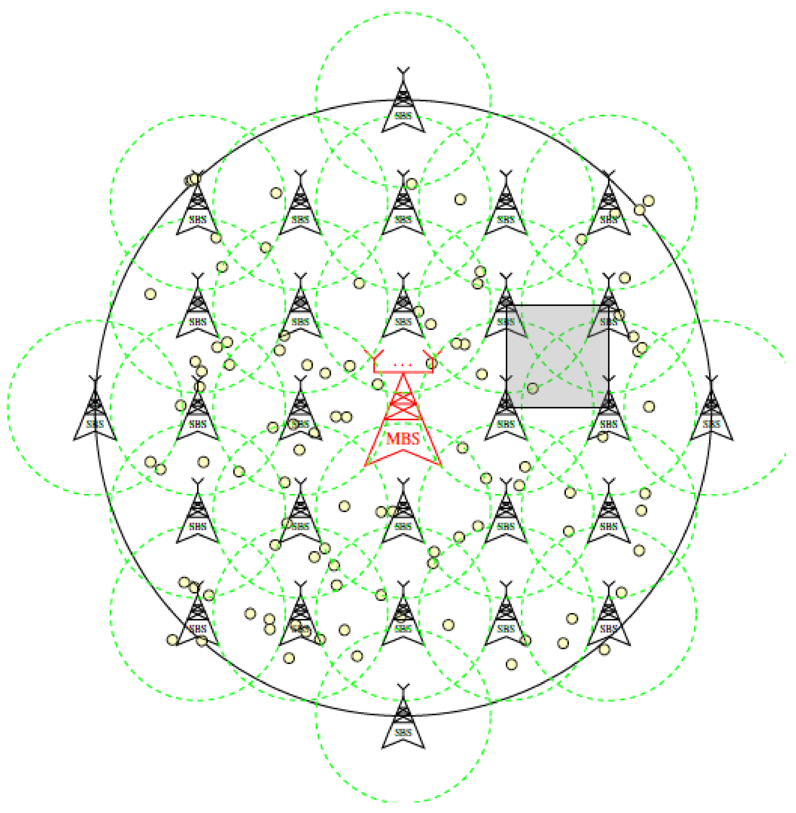
Mathematics and Algorithmic Sciences Lab
Huawei technologies Co. Ltd., France
20 quai du point du jour, 92100 Boulogne-Billancourt

HUAWEI

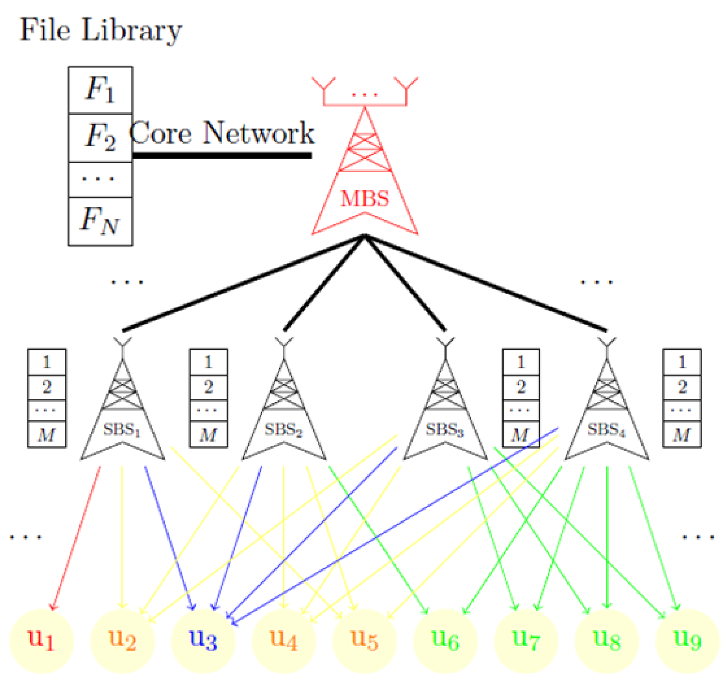
September 7, 2015 | Seminar on modeling, optimization, and control in wireless networks | Telecom ParisTech, Paris

Research Problem

We investigate the problem of optimal MDS-encoded cache placement at the wireless edge to minimize the backhaul rate in heterogeneous networks.



- U wireless users send requests to download files from a macro-cell base station.
- The macro-cell base station (MBS) has access to a library of N files $\mathcal{F} = \{F_1, \dots, F_N\}$, each of size B bits. File F_j is requested with probability p_j with $\sum_{j=1}^{N} p_j = 1$.
- N_{SBS} small-cell base stations with cache of size $M \cdot B$ bits are deployed to serve user requests within overage range r.
- Each user requesting for files in $\mathcal{F}$ is initially served by d_u SBSs depending on its location in the area, in particular, we call γ_i the probability for a user to be served by $d_u = i$ SBSs.
- If the requested file is not completely present in the caches, the MBS has to send the missing data to the user, using the backhaul connection.



MDS Coded Caching Scheme: Definition

1) Placement phase: The MBS creates $E_j = n + (N_{SBS} - 1)m_j$ encoded packets using a MDS(E_j, n) code. The MBS keeps $n - m_j$ encoded packets, and sends the other ones so that each SBS stores m_j unique packets.

2) Delivery phase: A user requesting file F_j contacts $d_u \geq 1$ SBSs, receiving $m_j d_u$ different encoded packets. If $m_j d_u \geq n$, the user can recover the file. Otherwise the MBS sends the remaining $n - m_j d_u \leq n - m_j$ encoded packets. Since the MBS kept $n - m_j$ encoded packets, the user does not receive replicated packets, and can decode the file.

Performance Analysis- Main Results

- MDS coded caching

Proposition The average backhaul rate for an encoded caching placement scheme $\mathcal{C}_{\mathbf{m}}^{\text{MDS}}$ defined by the placement $\mathbf{m} = [m_1 \cdots m_N]$ can be calculated as

$$R_{(\mathcal{C}_{\mathbf{m}}^{\text{MDS}})} = \sum_{d=1}^{S} \sum_{j=1}^{N} \gamma_i p_j \left(1 - \min\left(1, \frac{d m_j}{n}\right)\right), \qquad (1)$$

where $S \leq N_{\text{SBS}}$ is the maximum number of SBSs serving a user.

Idea of Proof Let u be a user served by d_u SBSs. Each SBS stores m_j packets of the file F_j, hence the user can collect $d_u m_j$ different encoded packets. If $d_u m_j \geq n$ the MBS does not send any packet, otherwise $n - d_u m_j$ packets are sent through the backhaul. The rate for user u requesting for file F_j can consequently be calculated as

$$R(F_j) = 1 - \min\left(1, \frac{d_u m_j}{n}\right). \qquad (2)$$

A user has the probability γ_d to be served by d SBSs and each file has a different probability to be requested by a user. Averaging over the request probability distribution $p = [p_1 \cdots p_N]$:

$$R_{(\mathcal{C}_{\mathbf{m}}^{\text{MDS}})} = \sum_{d=1}^{S} \sum_{j=1}^{N} \gamma_d p_j \left(1 - \min\left(1, \frac{d m_j}{n}\right)\right).$$

- Random uncoded caching

To evaluate the gain given by the use of MDS codes, we calculate the backhaul rate of a random caching scheme, where only file fragmentation is exploited.

Proposition The average backhaul rate for a random caching placement scheme $\mathcal{C}_{\mathbf{m}}$ defined by the placement $\mathbf{m} = [m_1 \cdots m_N]$ can be calculated as

$$R_{(\mathcal{C}_{\mathbf{m}})} = \sum_{d=1}^{S} \sum_{j=1}^{N} \gamma_i p_j \left(1 - \frac{m_j}{n}\right)^d. \qquad (3)$$

Given a cache placement $\mathbf{m} = [m_1 \cdots m_N]$, it is preferable in terms of backhaul usage to store MDS coded packets rather than fragments, in particular:

Proposition For any placement $\mathbf{m} = [m_1 \cdots m_N]$, it holds that

$$R_{(\mathcal{C}_{\mathbf{m}}^{\text{MDS}})} \leq R_{(\mathcal{C}_{\mathbf{m}})}.$$

Optimal MDS Coded Caching

Proposition Finding the optimal MDS coded placement scheme $\mathcal{C}_{(\text{opt})}^{\text{MDS}}$ defined by $\mathbf{m}_{(\text{opt})} = [m_1 \cdots m_N]$, which minimizes the average backhaul rate $R_{(\mathcal{C}_{\mathbf{m}}^{\text{MDS}})}$, is a convex optimization problem:

$$\begin{aligned} \min_{q_1,\dots,q_N} & \sum_{d=1}^{S} \sum_{j=1}^{N} \gamma_d p_j (1 - \min(1, d q_j)) \\ \text{s.t. } & \sum_{j=1}^{N} q_j = M \\ & 0 \leq q_j \leq 1 \qquad \forall j \in [1, N] \end{aligned} \qquad (4)$$

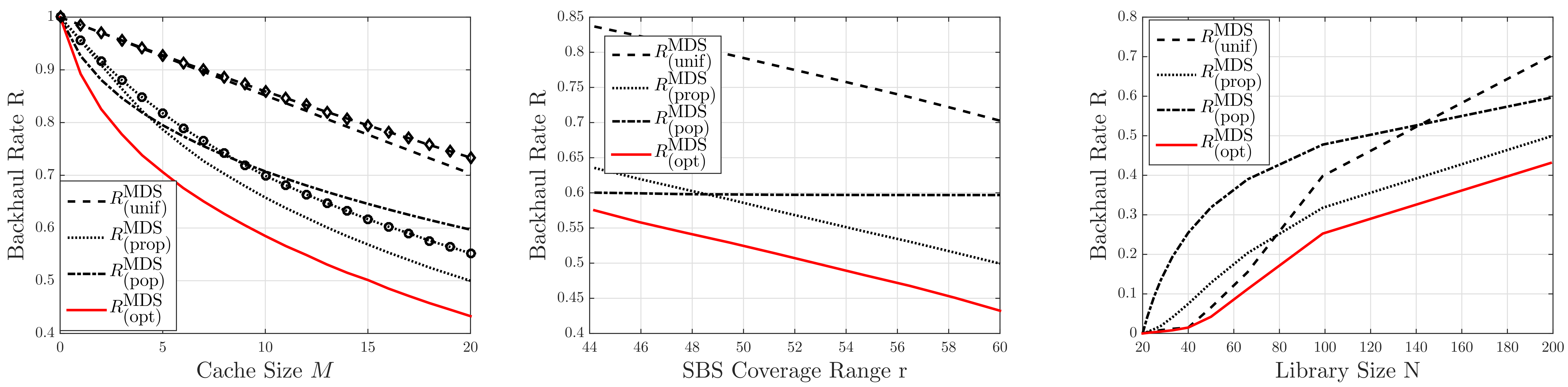
where $q_j = m_j / n$.

Further details

- To be presented at IEEE Globecom 2015, December, San Diego, U.S.A.
- Preprint available at http://arxiv.org/abs/1508.05753

Numerical Illustrations

SBSs are deployed according to a regular grid with a distance $d = 60$ meters between each SBS. The MBS coverage area $\mathcal{R}$ has a radius of $D = 500$ meters. Each SBS has a coverage area of radius r such that $d/\sqrt{2} \leq r \leq d$. We consider the users to be uniformly distributed in $\mathcal{R}$, with density $\rho_d = \rho = 0.05$ users/m^2. These numbers correspond to 316 SBSs being deployed, covering $U = 31,415$ mobile users. The request probability of the files $p = [p_1 \cdots p_N]$ is distributed according to a Zipf law of parameter $\alpha = 0.7$, i.e.,$p_j = \frac{1/j^\alpha}{\sum_j 1/j^\alpha}$.



We compare the optimal achievable backhaul rate $R_{(\text{opt})}^{\text{MDS}}$ to three other practical placement schemes, with and without MDS encoding of the files:

- "most popular" placement $R_{(\text{pop})}$ and $R_{(\text{pop})}^{\text{MDS}}$: each SBS stores the M most popular files;
- uniform placement $R_{(\text{unif})}$ and $R_{(\text{unif})}^{\text{MDS}}$: the file is divided into n fragments, which are placed uniformly at random at each SBS, i.e., each SBS stores M/N fragments of each file;
- proportional placement $R_{(\text{prop})}$ and $R_{(\text{prop})}^{\text{MDS}}$: for each file F_j, n fragments are created. The amount of parts m_j of file F_j stored at the SBSs is proportional to p_j while satisfying the size constraints.