Resource allocation on highly-distributed content delivery networks

Juan Vanerio (juan.vanerio@univie.ac.at - University of Vienna, Austria)
Stefan Schmid (stefan.schmid@tu-berlin.de - TU Berlin, Germany)

Introduction

Driven by the ever-increasing traffic demand for content and low latencies, storage resources are being deployed closer to the end users. From information-centric networks to 5G wireless systems, caching popular content close to the network edge can alleviate performance bottlenecks and enhance the end-user experience. This potential led to the proliferation of Content Delivery Networks (CDNs), a component of the Internet ecosystem that manages distributed caches.

Although classical caching techniques provide worst-case guarantees (e.g. Least-Recently-Used and its variants), current research has focused on finding sophisticated techniques achieving good performance on seen requests. We focus on the following (sub)-problems:

1. which items to cache at each device (**content allocation**); and

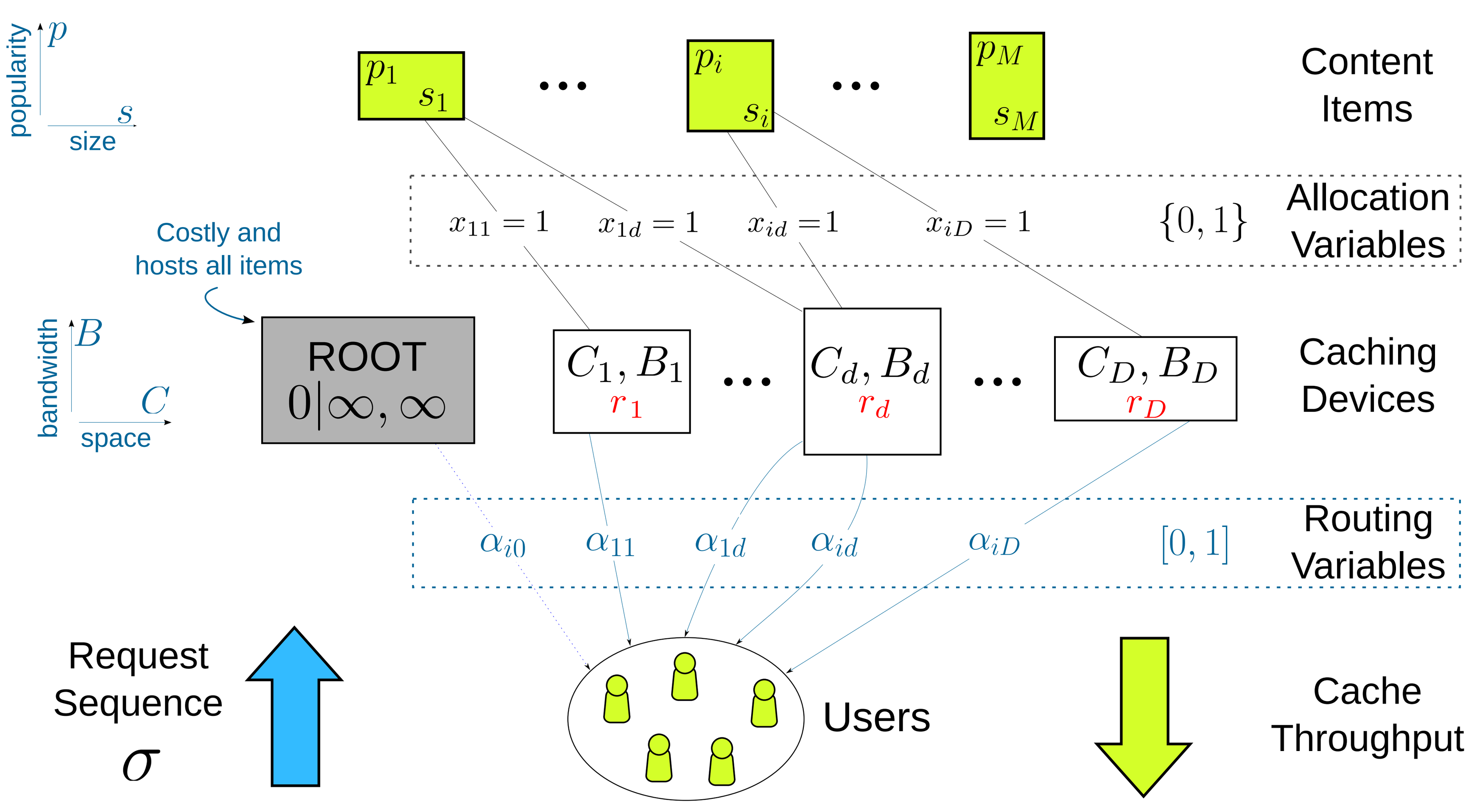
2. how to route content from devices to end-users (**routing policy**),

while considering that end-users are not spread too thin among caching devices. The overall problem is NP-Hard to solve and to approximate. Therefore it is a good candidate to be addressed with Machine Learning techniques.

Project Objective

- Decrease ROOT server traffic with respect to realizable baselines.
- Ability to make fast decisions.
- Leverage item popularity prediction.
- Provide quality of service to users.

Model



- N_i = Requests to item i up to time T.
- r_d: concurrent requests at d. Not observable!

NP-HARD!

- $\max_{\mathbf{x},\alpha} \sum_{i=1}^{M} s_i N_i (1-\alpha_{i0})$
- Subject to:
  - **Storage capacity**: $\sum_{i=1}^{M} x_{id} s_i \le C_d \quad \forall d \in [D]$
  - **Bandwidth**: $\sum_{i=1}^{M} s_i N_i \alpha_{id} \le B_d \quad \forall d \in [D]$
  - **Admission**: $r_d \le R_d \quad \forall d \in [D]$
  - **Demand conservation**: $\sum_{d=0}^{D} \alpha_{id} = 1 \quad \forall i \in [M]$
  - $0 \le \alpha_{id} \le x_{id} \le 1 \quad \forall i \in [M], d \in [D]$

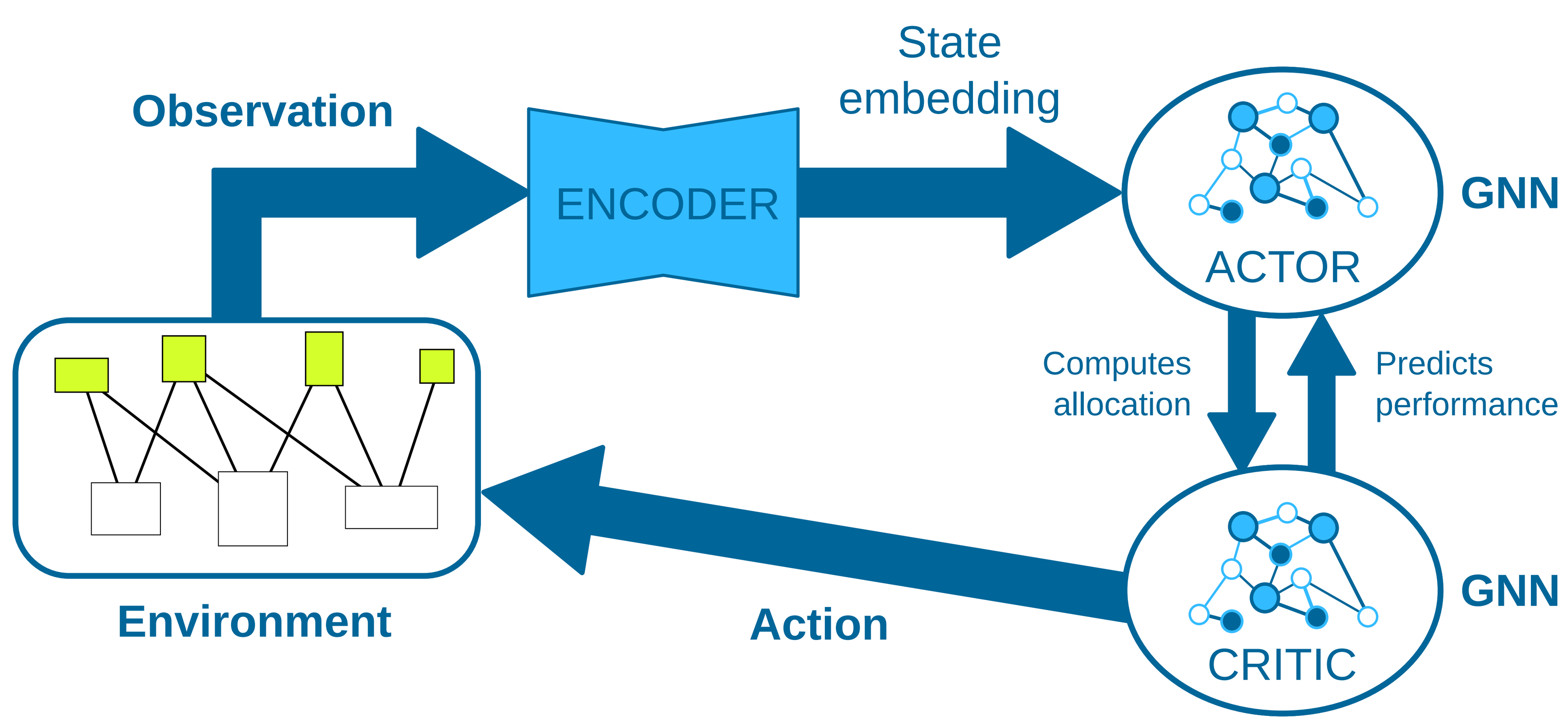
Possible Machine Learning Approaches

Static setting

- Constrained search algorithm for allocation, π-routing.

Slotted time (piece-wise static)

Reinforcement Learning for long-run changing allocations, π-routing.



- **State**: item and devices characteristics + allocation matrix $\mathbf{A}$.
- **Reward**: Based on transmission stats.
  - Real: tx-bytes minus prefetching costs.
  - Estimate (**CRITIC**): GNN trained on bipartite graph.
  - Baseline: flow-based linear approximation.
- **Actions**:
  - Choose $\mathbf{A}$ for the next time slot.
    - Baseline: heuristic composition.
  - **ACTOR** Idea **1**: Link prediction GNN.
  - **ACTOR** Idea **2**: Online RL- sequential item allocation.
    - Can accomodate changing item sets.

Dynamic (Future Research!)

Time-slotted or fully dynamic allocation, dynamic routing.

- Idea **1**: Augment devices state with remaining bytes for each request.
- Idea **2**: Compute a new graph with requests as nodes.

How would you approach the problem?
Willing to explore new ideas!

Baseline

Heuristics:

1. For given $\mathbf{X}$, optimal routing minimizes traffic from ROOT.

2. Allocation: Place item with largest remaining value on device with largest remaining (bandwidth to space) ratio. Repeat.

3. Routing: (π) send request to the estimated argmin(requests) device.

References