Performance and Scalability of Storage Systems
Per3s 2025

Bridging Local Efficiency and Global Cost: Two Complementary ICN Caching Strategies

Lydia Ait Oucheggou, Stéphane Rubini, Abdella Battou, Jalil Boukhobza

NIST, Gaithersburg, MD, USA
Univ. Bretagne Occidentale, Lab-STICC, CNRS, UMR 6285, F-29200 Brest, France
Lab-STICC, CNRS UMR 6285, ENSTA, Institut Polytechnique de Paris, 29806 Brest, France

23 Mai 2025

Outline

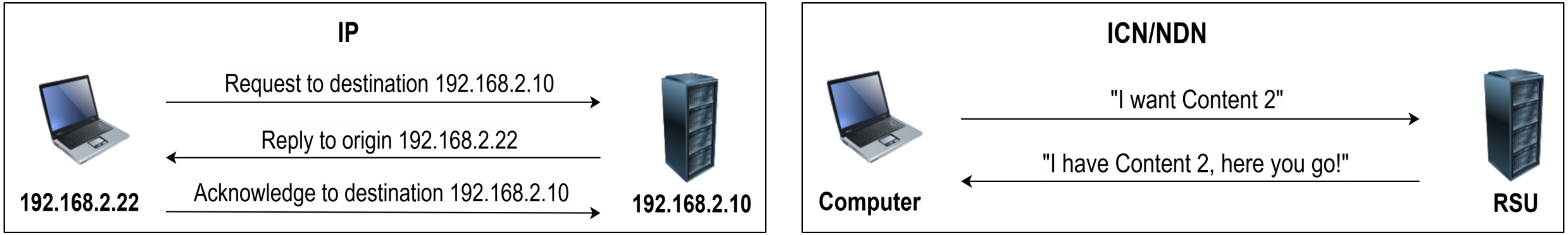
1. Context
2. Problems Statements
3. Solution
4. Evaluation
5. Conclusion and Perspective

Outline

Context
Data Explosion and the Shift to ICN

- 149 ZB of data created in 2024 ➜ Projected 394 ZB by 2028 [1].
- Traditional IP networks struggle with Scalability and Efficiency [2, 3].
- ICN (Information-Centric Networking) [4] addresses this via:
  - Content-based addressing **&** In-network caching

→The need for **caching strategies** to improve performance & scalability



Same references as in the paper

Context

Caching Challenges: Node and Network Perspectives

Node-side (Centralized Strategies):

- Storage demand grows faster than deployment capacity [7].
- Multi-tier architectures (DRAM, SSD, HDD) complicate caching due to varying costs, lifespans, and performance [8-10].
- User-specific Service Level Agreements - SLAs define performance expectations [4], but most caching approaches ignore per-user QoS [5].

Network-side (Distributed Strategies):

- High energy and bandwidth costs associated with data movement [11].
- QoS constraints (latency, throughput …) via SLAs.
- Caching strategies overlook QoS, limiting overall efficiency.

Same references as in the paper

Outline

1. Context
2. Problems Statements
3. Solution
4. Evaluation
5. Conclusion and Perspective

Problems Statements

1. How do we design a multi-tier cache for applications that require different quality of service ?

2. How can we design a distributed caching strategy that reduces redundancy and cost?

State of the art

Why Current Caching Falls Short: Node and Network Perspectives

Category	Centralized				Distributed			
Criteria	Recency	Frequency	QoS	Heterogeneity	Cost	Redundancy	Popularity	Energy
LRU	✓	✗	✗	✗	✗	✗	✗	✗
LFU	✗	✓	✗	✗	✗	✗	✗	✗
LIRS [12], ARC [13], LeCar [26], LHD [14],autocache [27], CALC [15], ML-LIRS [28],Cacheus [16], SS-LRU [17], GL-Cache [29],Baleen [18]	✓	✓	✗	✗	✗	✗	✗	✗
Flashield [30]	✓	✓	✗	✓	✗	✗	✗	✗
QM-ARC [19]	✓	✓	✓	✓	✗	✗	✗	✗
LCE/LCD [20]	✗	✗	✗	✗	✗	✓	✓	✗
MPC [31], MAGIC [32], CPCCS [23],PDPU [24], CPCache [25]	✗	✗	✗	✗	✗	✓	✓	✗
CL2SM	✓	✓	✓	✓	✓	✓	✓	✓

Same references as in the paper

UBO Université de Bretagne Occidentale · Lab-STICC · NIST · ENSTA · INSTITUT POLYTECHNIQUE DE PARIS

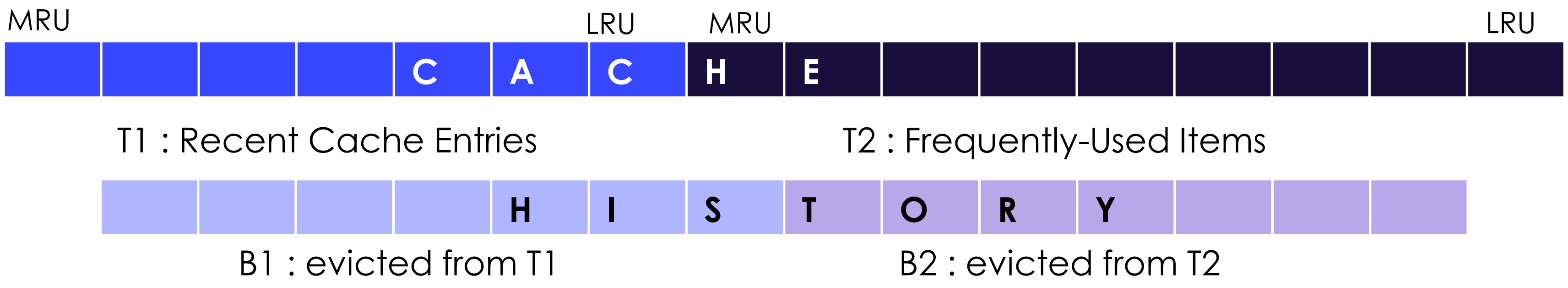
Outline

1. Context
2. Problem Statement
3. Solutions
4. Evaluation
5. Conclusion and Perspective

Context

ARC - Adaptive Cache Replacement

- **ARC - Adaptive Cache Replacement [6]:**
  - One of the most popular and efficient generic algorithms in the rich caching literature.
  - ARC maintains two LRU lists: T1 contains objects that have only been requested once, and T2 contains objects that have been requested at least twice.
  - ARC uses two LRU ghost lists: B1 and B2, which contain references to data that has been evicted from T1 and T2, respectively.
  - Cache miss → Insert in Most Recently Used-MRU position in T1.
  - Hit in T1, T2, B1, B2 → Promote data to MRU position in T2.
  - ARC dynamically adapts P, the size of the list T1 (size of T2 = cache size - P). Hit in B1 → Increase in P. Hit in B2 → decrease in P.
  - ARC does not take QoS into account and is originally designed for single-tier cache.

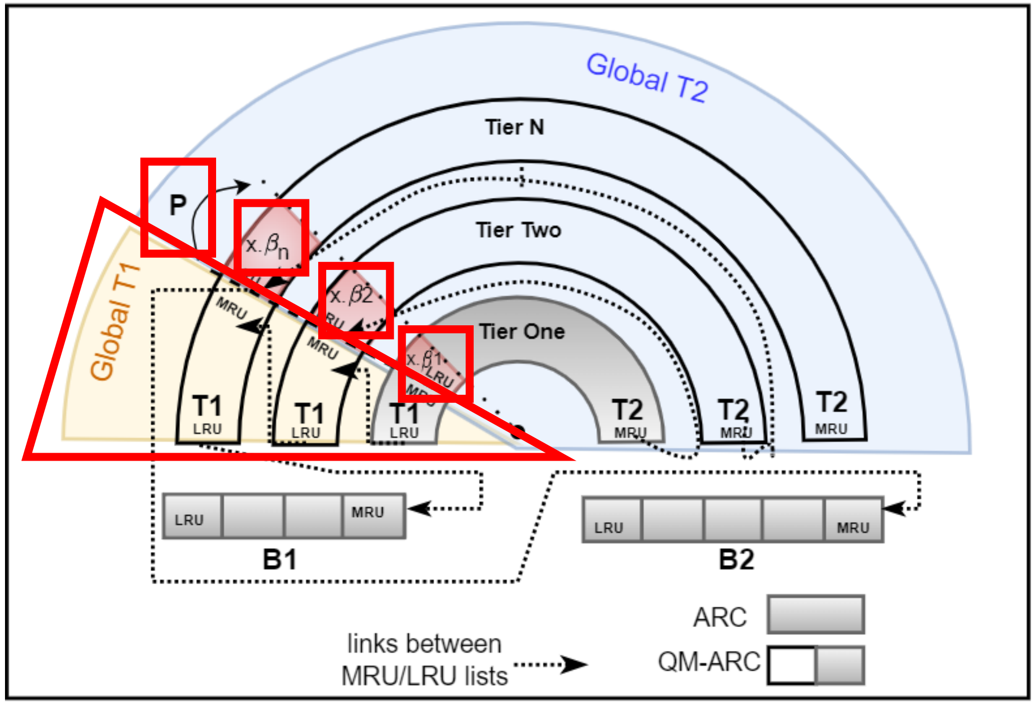


Same references as in the paper

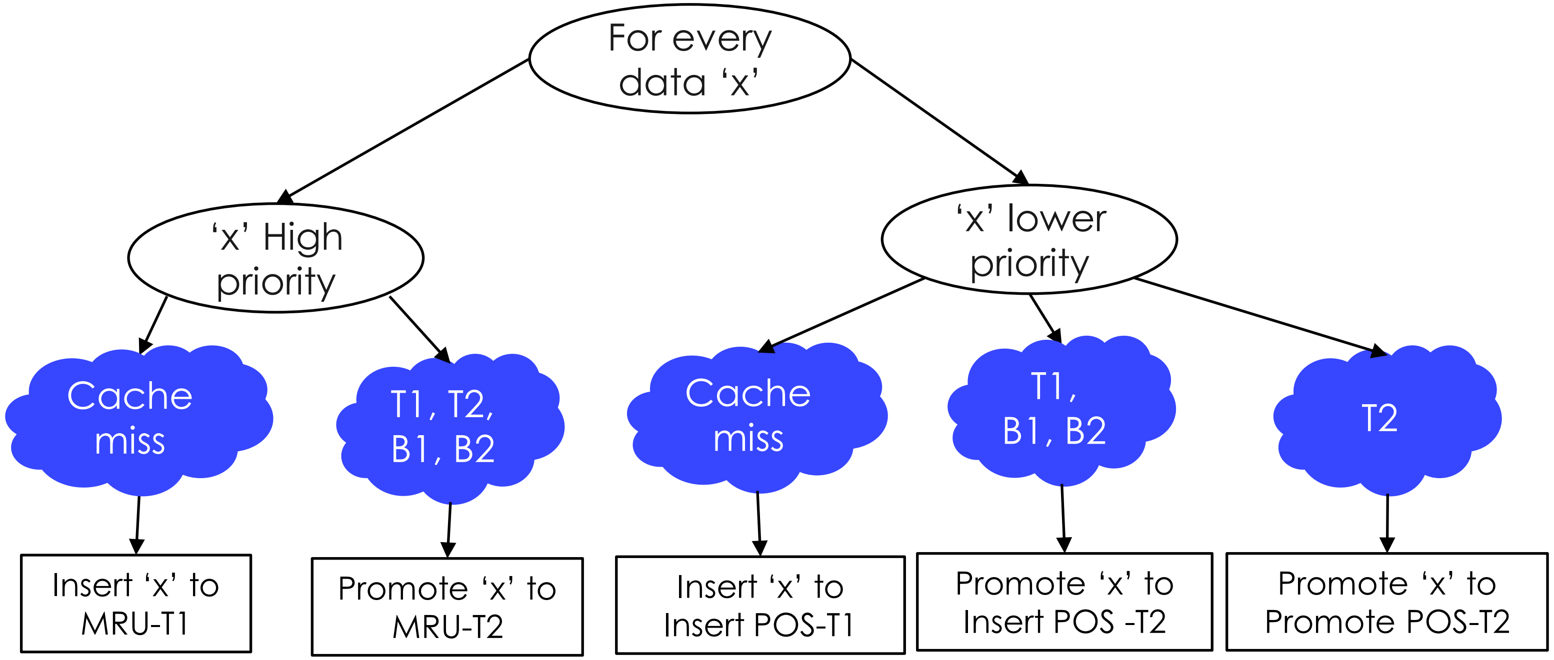
Université de Bretagne Occidentale

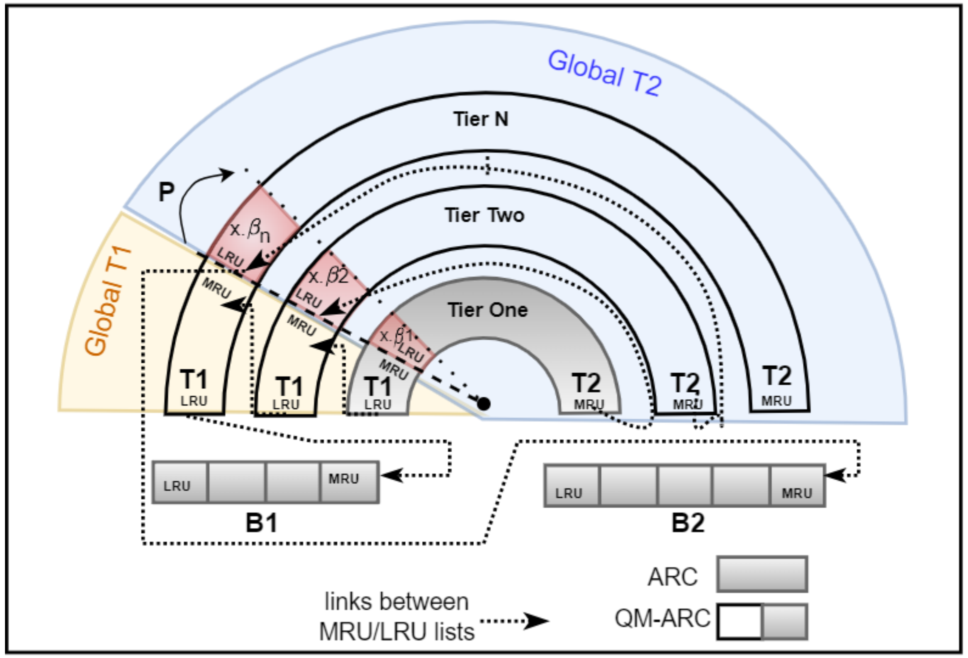
QM-ARC
QoS-aware, Multi-tier ARC policy

- QM-ARC improves upon ARC in two key areas:
  - **Multi-Tier Caches**: QM-ARC supports heterogeneous memory systems through dynamic local and global cache list size adjustments across the tiers using β_i.
  - **QoS-Based Caching**: QM-ARC integrates QoS by adopting SLA and penalty-based management, from the Cloud, using priority levels to calculate the index for insertions and promotions using γ_k.



QM-ARC
Explained





- Index calculation for insertion and promotion based on QoS using $\gamma_k = \mathrm{Penalty}_k \ / \ \mathrm{Penalty}_0$
- Proportional size adjustment of the global lists, distributed between tiers using $\beta_i = |\mathrm{tier}_i| \ / \ |\mathrm{tier}_1|$

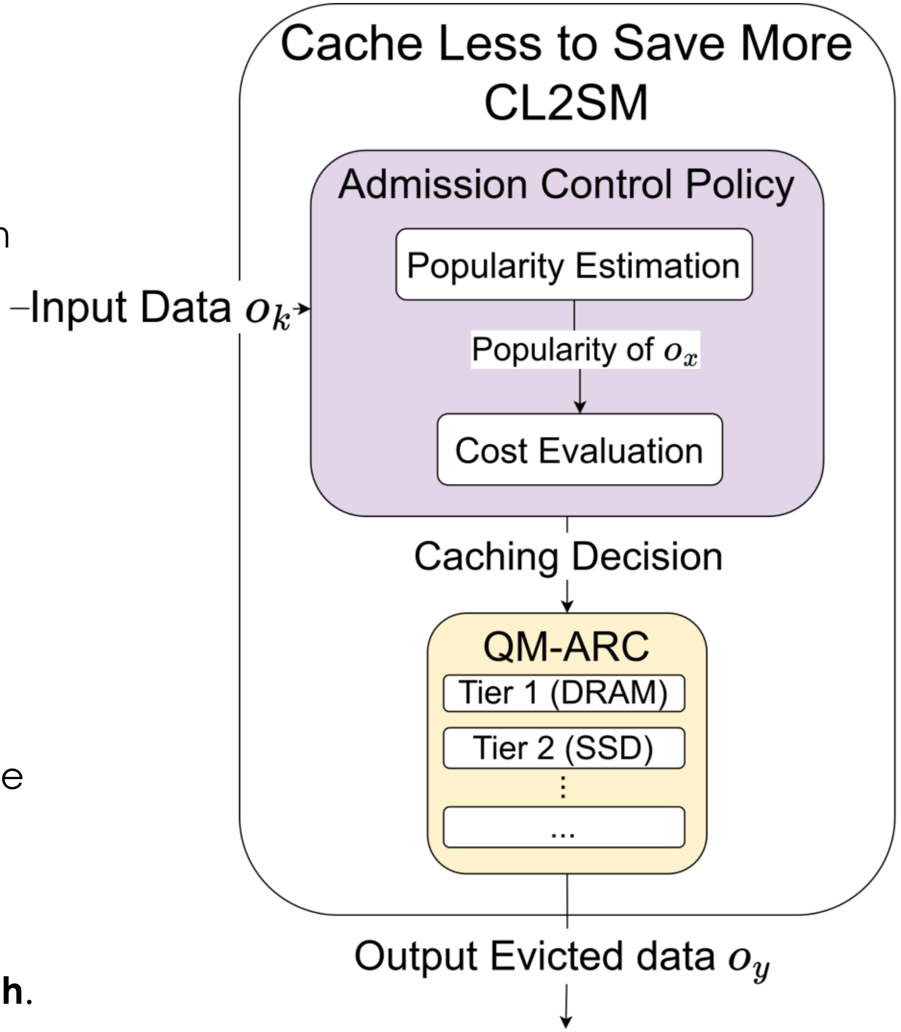
CL2SM
Cache Less to Save More

- Our proposed CL2SM caching strategy is composed of **three** core modules:

1. **Popularity Estimation:**
   - Each node maintains per-item request counters that are incremented with every request (even if not cached)
   - Popularity is calculated only for cached items:

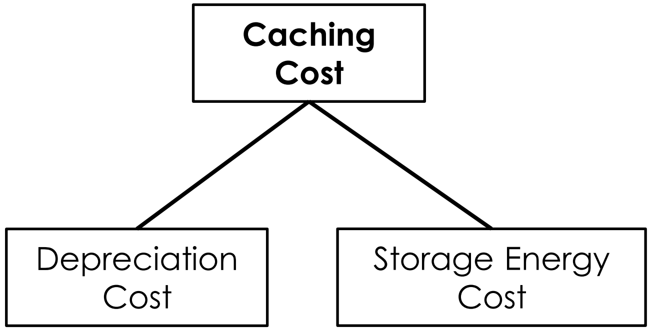
$$\text{Popularity(k)} = \frac{\text{Requests for data k}}{\text{Requests for all data in current node}}$$

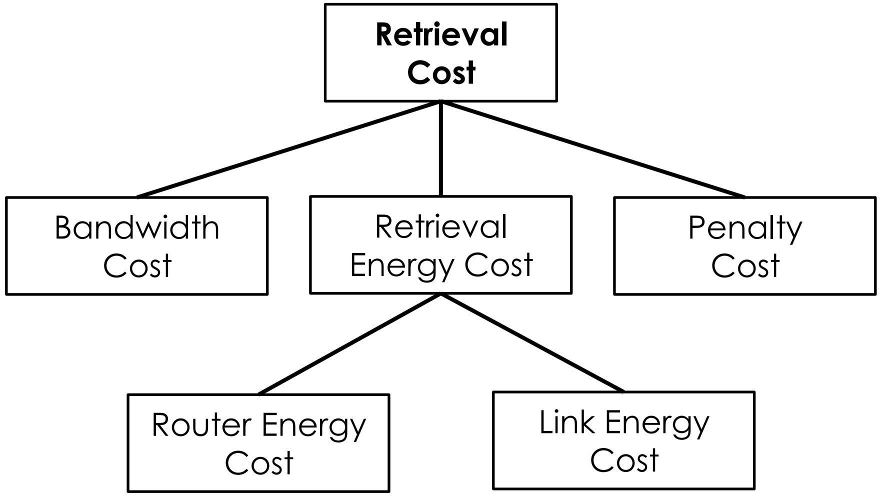
2. **Cost Evaluation:**
   - Determines the benefit of caching each item based on a **Gain vs. Loss** principle:
     - **Gain** = (Local Popularity) × (**Retrieval Cost** Saved by Caching)
     - **Loss** = (**Caching Cost** of the Item) + (Cost of Evicting an Existing Item)
   - If **Gain > Loss**, the item with the lowest gain in the cache is evicted to make space.
3. **Enhanced QM-ARC:**
   - Evicts data based on both **recency** and **retrieval cost**.
   - From the least recently used 10%, we remove the **cheapest item to re-fetch**.



CL2SM
Cost Model

- **Caching cost**: The cost incurred when storing data locally, including device depreciation and the energy consumed during read and write operations.
- **Retrieval cost**: The cost of fetching data from remote nodes when it is not cached locally. This includes band-width usage, energy for routing and forwarding, and potential penalties for SLA violation.

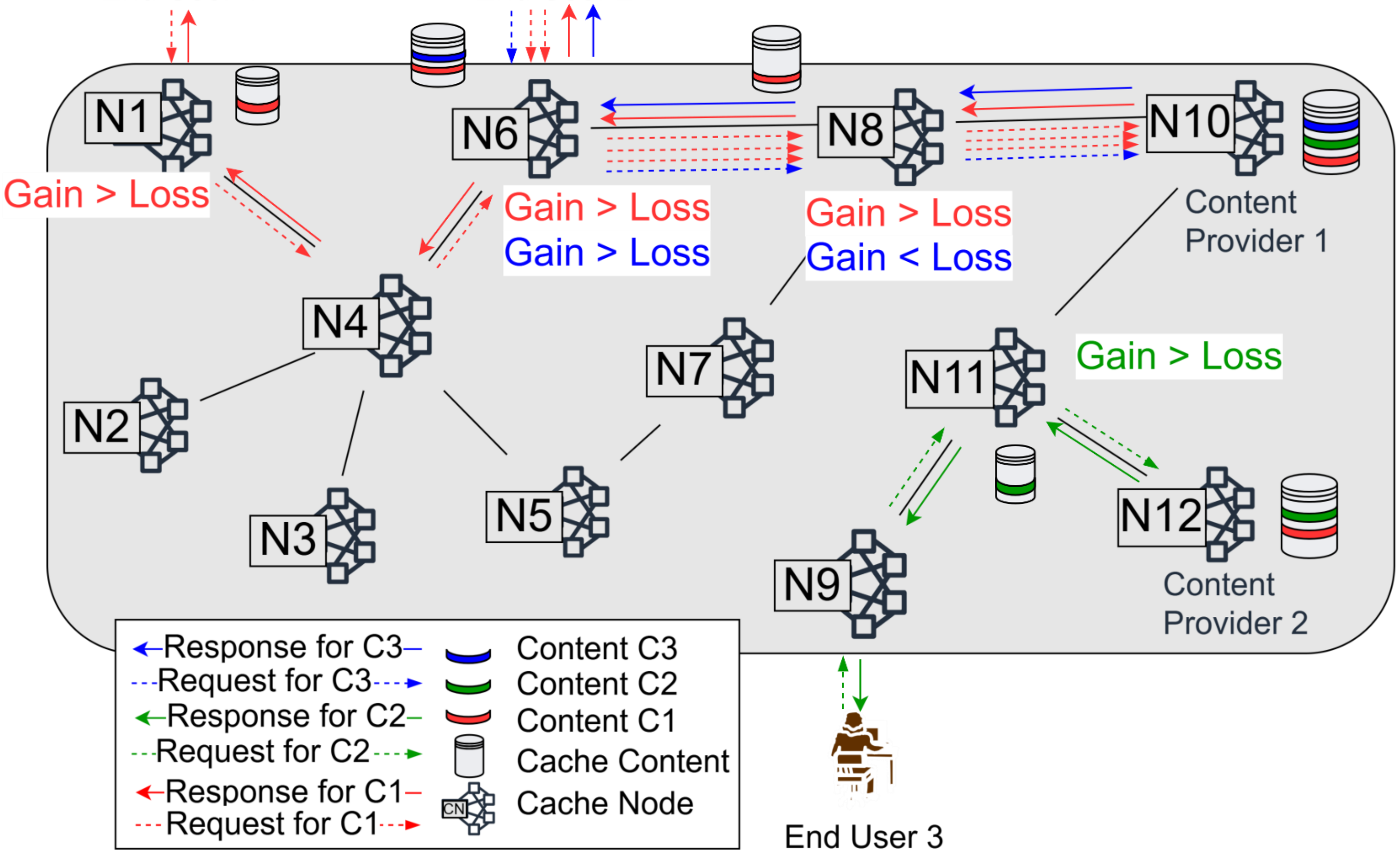




CL2SM
Recapitulative

ICN Architecture

Cache Node N8		
Content Name	Request Counter	Cache Gain
C1	3	0.9
C3	1	0.2



Université de Bretagne Occidentale

Outline

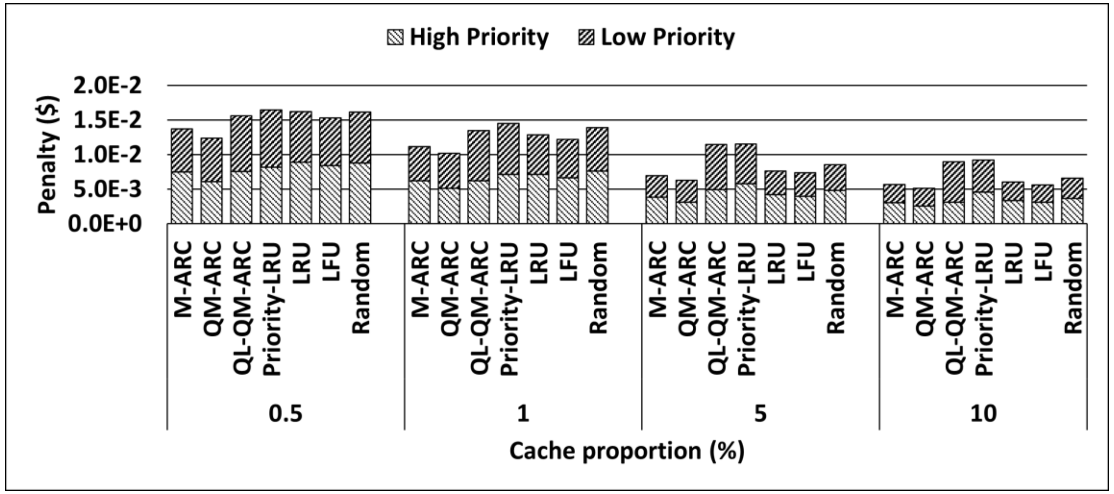
1. Context
2. Problem Statement
3. Solution
4. Evaluation
5. Conclusion and Perspective

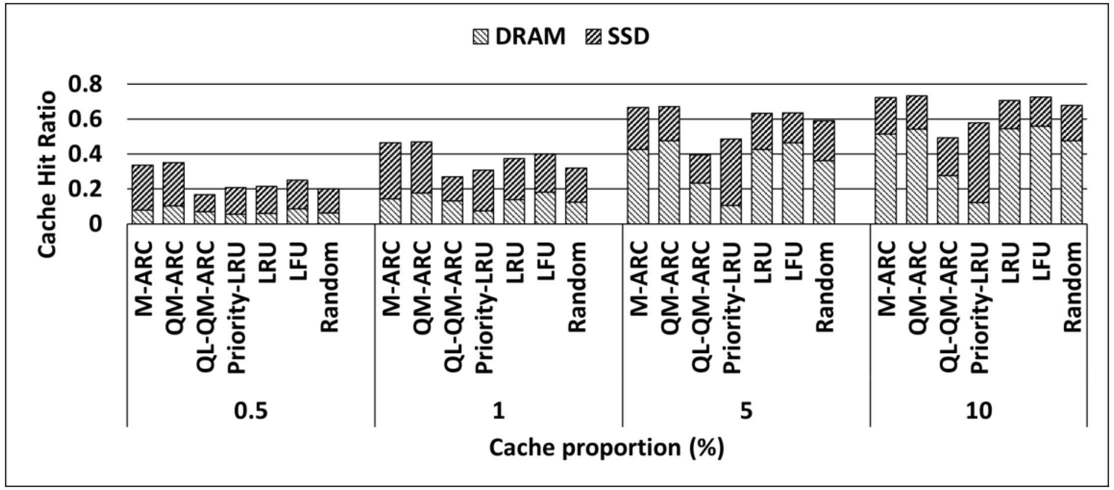
Evaluation : QM-ARC

- **Simulator:**
  - Implementation of a cache simulator with Simpy a discrete event simulation framework based on standard Python.
  - Link to the simulator: https://github.com/Multi-Tier-Cache-Simulator/MultiTierCacheSimulator

- **Methodology:**
  - **Experiment**: vary cache proportion as a function of dataset size [0.5% - 10%].
  - **Traces**: use of Zif-like synthetic and real traces (IBM and Jedi [28]).
  - **Solutions compared**: **QM-ARC** and LRU, LFU, Random, Priority-LRU[16], M-ARC.
  - **Evaluation criteria**: penalty, overall hit rate, hit rate per priority level
  - **Two** priority levels: **20%** of the data is high priority.
  - **System**: 2-tiers, the first one is DRAM, the second is SSD, the size of the DRAM is a fifth of the SSD

Results: QM-ARC





- **Penalty:**
  - QM-ARC cuts penalty by 80% as cache size grows.
- **Global Hit Ratio:**
  - M-ARC & QM-ARC lead with a 48% increase.
- **High-Priority Hit Rate:**
  - QM-ARC boosts hits by 67%, leveraging the fastest tier.
- **Low-Priority Hit Rate:**
  - M-ARC outperforms as QM-ARC favors high-priority data—validating its QoS strategy.

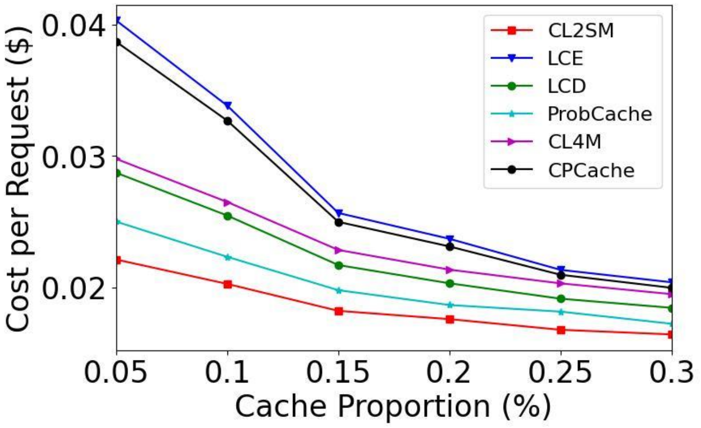
Université de Bretagne Occidentale

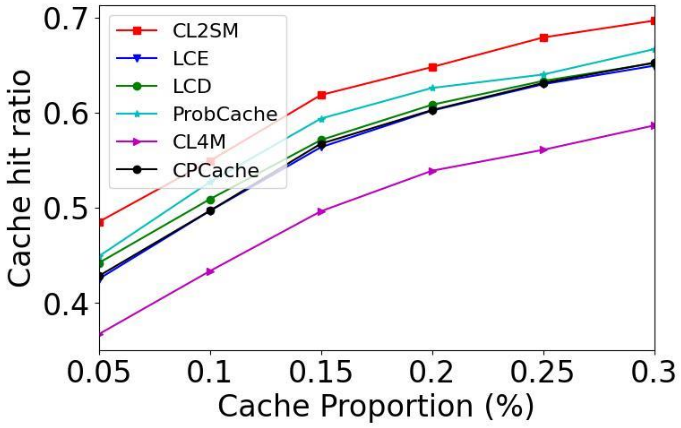
Evaluation : CL2SM

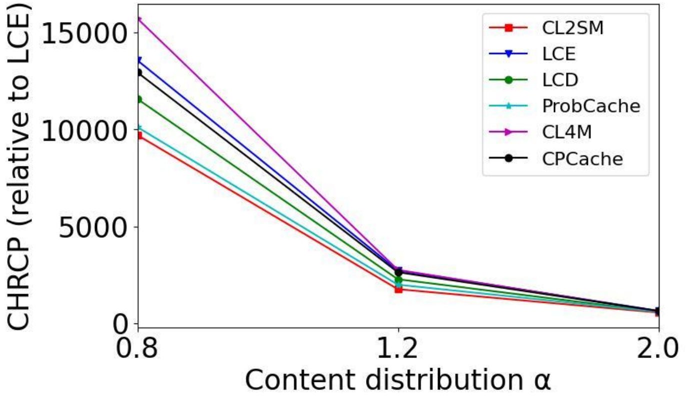
- **Simulator**:
  - Icarus[37] a Python-based discrete-event simulator developed for ICN, focusing on the performance evaluation of caching mechanisms.
  - Link to the simulator: https://github.com/LydiaNosali/icarus/tree/qmarc

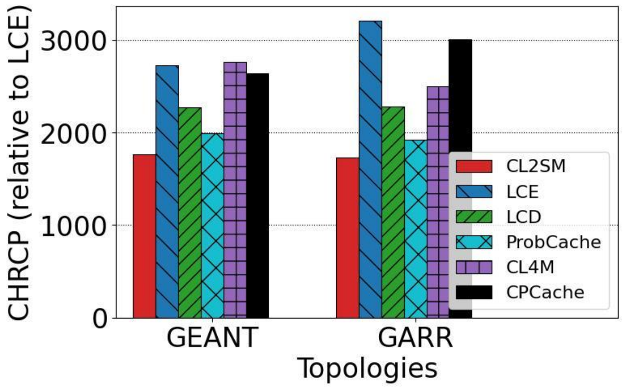
- **Methodology**:
  - **Experiments:**
    - **Experiment 1:** vary cache proportion as a function of dataset size [0.05% - 0.3%].
    - **Experiment 2:** vary content distribution alpha of the Zipf Law [0.8, 1.2, 2.0]
    - **Experiment 3:** vary topologies [GEANT and GARR].
  - **Solutions compared: CL2SM** and LCE, LCD, Prob-Cache [21], CL4M [24], CPCache [25].
  - **Evaluation criteria:** Cost per request, Global hit rate, Cache Hit Ratio Cost Product (CHRCP), Latency
  - **Two** priority levels: **20%** of the data is high priority.
  - **System**: 2-tiers, the first one is DRAM, the second is SSD, the size of the DRAM is a fifth of the SSD

Results: CL2SM









- CL2SM balances caching and retrieval costs, offering a trade-off between storage overhead and access efficiency.
- Caching more does not necessarily mean better performance.
- Strong performance at moderate skew ($\alpha = 1.2$), where both **cost** and **popularity** are relevant.
- **CL2SM** adapts effectively to diverse network structures as CHRCP is stable.

Outline

Conclusion and Perspective

Summary:

- Design of **CL2SM**, a unified caching algorithm for ICN that:
  - Integrates **centralized** and **distributed** caching strategies
  - Uses **content popularity** + **cost model** (energy, transmission, SLA penalties)
- Supports **multi-tier storage** with the use of **QM-ARC** as a replacement strategy

Benefits:

- **Adaptive & cost-efficient**
- **QoS-aware**, models real device behavior
- Promotes **performance & sustainability**

For future work:

- Reduce **carbon footprint** of caching

Thank you for your attention !

References

[1] T. Petroc, “Volume of data/information created, captured, copied, andconsumed worldwide from 2010 to 2023, with forecasts from 2024 to2028. tech. rep,” 2024.
[2] S. A. Mohammed and A. L. Ralescu, “Future internet architectures onan emerging scale—a systematic review,” Future Internet, vol. 15, no. 5,p. 166, 2023.
[3] A. Anjum, P. Agbaje, A. Mitra, E. Oseghale, E. Nwafor, and H. Olu-fowobi, “Towards named data networking technology: Emerging ap-plications, use cases, and challenges for secure data communication,”Future Generation Computer Systems, vol. 151, pp. 12–31, 2024.
[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman,“A survey of information-centric networking,” IEEE CommunicationsMagazine, vol. 50, no. 7, pp. 26–36, 2012.
[5] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez,T. Schmidt, and M. Waehlisch, “Rfc 7927: Information-centric network-ing (icn) research challenges,” URL https://tools. ietf. org/html/rfc7927,2016.
[6] G. White and G. Rutz, “Content delivery with content-centric network-ing,” CableLabs, Strategy & Innovation, pp. 1–26, 2016.
[7] Z. Tang, Y. Wang, X. He, L. Zhang, X. Pan, Q. Wang, R. Zeng, K. Zhao,S. Shi, B. He et al., “Fusionai: Decentralized training and deploying llmswith massive consumer-level gpus,” arXiv preprint arXiv:2309.01172,2023.
[8] M. Soltaniyeh, V. Lagrange Moutinho Dos Reis, M. Bryson, X. Yao,R. P. Martin, and S. Nagarakatte, “Near-storage processing for solid statedrive based recommendation inference with smartssds®,” in Proceedingsof the 2022 ACM/SPEC on International Conference on PerformanceEngineering, 2022, pp. 177–186.
[9] L. Zhang, R. Karimi, I. Ahmad, and Y. Vigfusson, “Optimal dataplacement for heterogeneous cache, memory, and storage systems,”Proceedings of the ACM on Measurement and Analysis of ComputingSystems, vol. 4, no. 1, pp. 1–27, 2020.
[10] J. Boukhobza, S. Rubini, R. Chen, and Z. Shao, “Emerging nvm:A survey on architectural integration and research challenges,” ACMTrans. Des. Autom. Electron. Syst., vol. 23, no. 2, nov 2017. [Online].Available: https://doi-org.ins2i.bib.cnrs.fr/10.1145/3131848
[11] J. Li, B. Liu, and H. Wu, “Energy-efficient in-network caching forcontent-centric networking,” IEEE Communications Letters, vol. 17,no. 4, pp. 797–800, 2013.
[12] S. Jiang and X. Zhang, “Lirs: An efficient low inter-reference recencyset replacement policy to improve buffer cache performance,” ACMSIGMETRICS Performance Evaluation Review, vol. 30, no. 1, pp. 31–42, 2002.
[13] N. Megiddo and D. S. Modha, “Arc: A self-tuning, low overheadreplacement cache.” in Fast, vol. 3, 2003, pp. 115–130.
[14] N. Beckmann, H. Chen, and A. Cidon, “{LHD}: Improving cachehit rate by maximizing hit density,” in 15th USENIX Symposium onNetworked Systems Design and Implementation (NSDI 18), 2018, pp.389–403.
[15] M. Kachmar and D. Kaeli, “Calc: A content-aware learning cache forstorage systems,” in 2021 IEEE International Conference on Network-ing, Architecture and Storage (NAS). IEEE, 2021, pp. 1–8.
[16] L. V. Rodriguez, F. Yusuf, S. Lyons, E. Paz, R. Rangaswami, J. Liu,M. Zhao, and G. Narasimhan, “Learning cache replacement with{CACHEUS},” in 19th USENIX Conference on File and Storage Tech-nologies (FAST 21), 2021, pp. 341–354.
[17] C. Li, M. Wu, Y. Liu, K. Zhou, J. Zhang, and Y. Sun, “Ss-lru: a smartsegmented lru caching,” in Proceedings of the 59th ACM/IEEE DesignAutomation Conference, 2022, pp. 397–402.
[18] D. L.-K. Wong, H. Wu, C. Molder, S. Gunasekar, J. Lu, S. Khand-kar, A. Sharma, D. S. Berger, N. Beckmann, and G. R. Ganger,“Baleen:{ML} admission & prefetching for flash caches,” in 22ndUSENIX Conference on File and Storage Technologies (FAST 24), 2024,pp. 347–371.
[19] L. Ait-Oucheggou, S. Rubini, A. Battou, and J. Boukhobza, “QM-ARC: Qos-aware multi-tier adaptive cache replacement strategy,” FutureGeneration Computer Systems, vol. 163, p. 107548, 2025.
[20] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs,and R. L. Braynard, “Networking named content,” in Proceedings of the5th international conference on Emerging networking experiments andtechnologies, 2009, pp. 1–12.

References

[21] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic in-network cachingfor information-centric networks,” in Proceedings of the second editionof the ICN workshop on Information-centric networking, 2012, pp. 55–60.
[22] W. K. Chai, D. He, I. Psaras, and G. Pavlou, “Cache “less for more” ininformation-centric networks (extended version),” Computer Communi-cations, vol. 36, no. 7, pp. 758–770, 2013.
[23] M. A. Naeem, S. A. Nor, S. Hassan, and B.-S. Kim, “Compound popularcontent caching strategy in named data networking,” Electronics, vol. 8,no. 7, p. 771, 2019.
[24] B. Nour, H. Khelifi, H. Moungla, R. Hussain, and N. Guizani, “Adistributed cache placement scheme for large-scale information-centricnetworking,” IEEE Network, vol. 34, no. 6, pp. 126–132, 2020.
[25] N. Hubballi and P. Chaudhary, “Cpcache: Cooperative popularity basedcaching for named data networks,” in 2024 International Conference onInformation Networking (ICOIN). IEEE, 2024, pp. 379–384.
[26] G. Vietri, L. V. Rodriguez, W. A. Martinez, S. Lyons, J. Liu, R. Ran-gaswami, M. Zhao, and G. Narasimhan, “Driving cache replacementwith {ML-based}{LeCaR},” in 10th USENIX Workshop on Hot Topicsin Storage and File Systems (HotStorage 18), 2018.
[27] H. Herodotou, “Autocache: Employing machine learning to automatecaching in distributed file systems,” in 2019 IEEE 35th internationalconference on data engineering workshops (ICDEW). IEEE, 2019, pp.133–139.
[28] R. Fabbro, C. Zhong, and S. Jiang, “Ml-lirs: Leveraging machine learn-ing to improve the lirs replacement algorithm,” in 2021 InternationalConference on High Performance Big Data and Intelligent Systems(HPBD&IS). IEEE, 2021, pp. 74–78.
[29] J. Yang, Z. Mao, Y. Yue, and K. Rashmi, “{GL-Cache}: Group-levellearning for efficient and high-performance caching,” in 21st USENIXConference on File and Storage Technologies (FAST 23), 2023, pp. 115–134.
[30] A. Eisenman, A. Cidon, E. Pergament, O. Haimovich, R. Stutsman,M. Alizadeh, and S. Katti, “Flashield: a hybrid key-value cache thatcontrols flash write amplification,” in 16th USENIX Symposium onNetworked Systems Design and Implementation (NSDI 19), 2019, pp.65–78.
[31] C. Bernardini, T. Silverston, and O. Festor, “Mpc: Popularity-basedcaching strategy for content centric networks,” in 2013 IEEE interna-tional conference on communications (ICC). IEEE, 2013, pp. 3619–3623.
[32] J. Ren, W. Qi, C. Westphal, J. Wang, K. Lu, S. Liu, and S. Wang,“Magic: A distributed max-gain in-network caching strategy ininformation-centric networks,” in 2014 IEEE conference on computercommunications workshops (INFOCOM WKSHPS). IEEE, 2014, pp.470–475.
[33] J. Iqbal, Z. u. Abideen, N. Ali, S. H. Khan, A. Rahim, A. Zahir, S. A. H.Mohsan, and M. H. Alsharif, “An energy efficient local popularitybased cooperative caching for mobile information centric networks,”Sustainability, vol. 14, no. 20, p. 13135, 2022.
[34] Y. Gu, Y. Li, H. Wang, L. Liu, K. Zhou, W. Fang, G. Hu, J. Liu,and Z. Cheng, “Lpca: learned mrc profiling based cache allocation forfile storage systems,” in Proceedings of the 59th ACM/IEEE DesignAutomation Conference, 2022, pp. 511–516.
[35] E. Ahvar, A.-C. Orgerie, and A. Lebre, “Estimating energy consumptionof cloud, fog, and edge computing infrastructures,” IEEE Transactionson Sustainable Computing, vol. 7, no. 2, pp. 277–288, 2019.
[36] A. Chikhaoui, L. Lemarchand, K. Boukhalfa, and J. Boukhobza, “Multi-objective optimization of data placement in a storage-as-a-service fed-erated cloud,” ACM Transactions on Storage (TOS), vol. 17, no. 3, pp.1–32, 2021.
[37] L. Saino, I. Psaras, and G. Pavlou, “Icarus: a caching simulator forinformation centric networking (icn),” in SimuTools, vol. 7. ICST,2014, pp. 66–75.
[38] G ´EANT, “G ´Eant network,” 2024, accessed: 2025-05-02. [Online].Available: https://network.geant.org/
[39] Consortium GARR, “Garr: The italian research and education network,”2025, accessed: 2025-05-02. [Online]. Available: https://www.garr.it/en/
[40] A. Chikhaoui, L. Lemarchand, K. Boukhalfa, and J. Boukhobza,“Stornir, a multi-objective replica placement strategy for cloud feder-ations,” in Proceedings of the 36th Annual ACM Symposium on AppliedComputing, 2021, pp. 50–59.

Université de Bretagne Occidentale

INSTITUT POLYTECHNIQUE DE PARIS