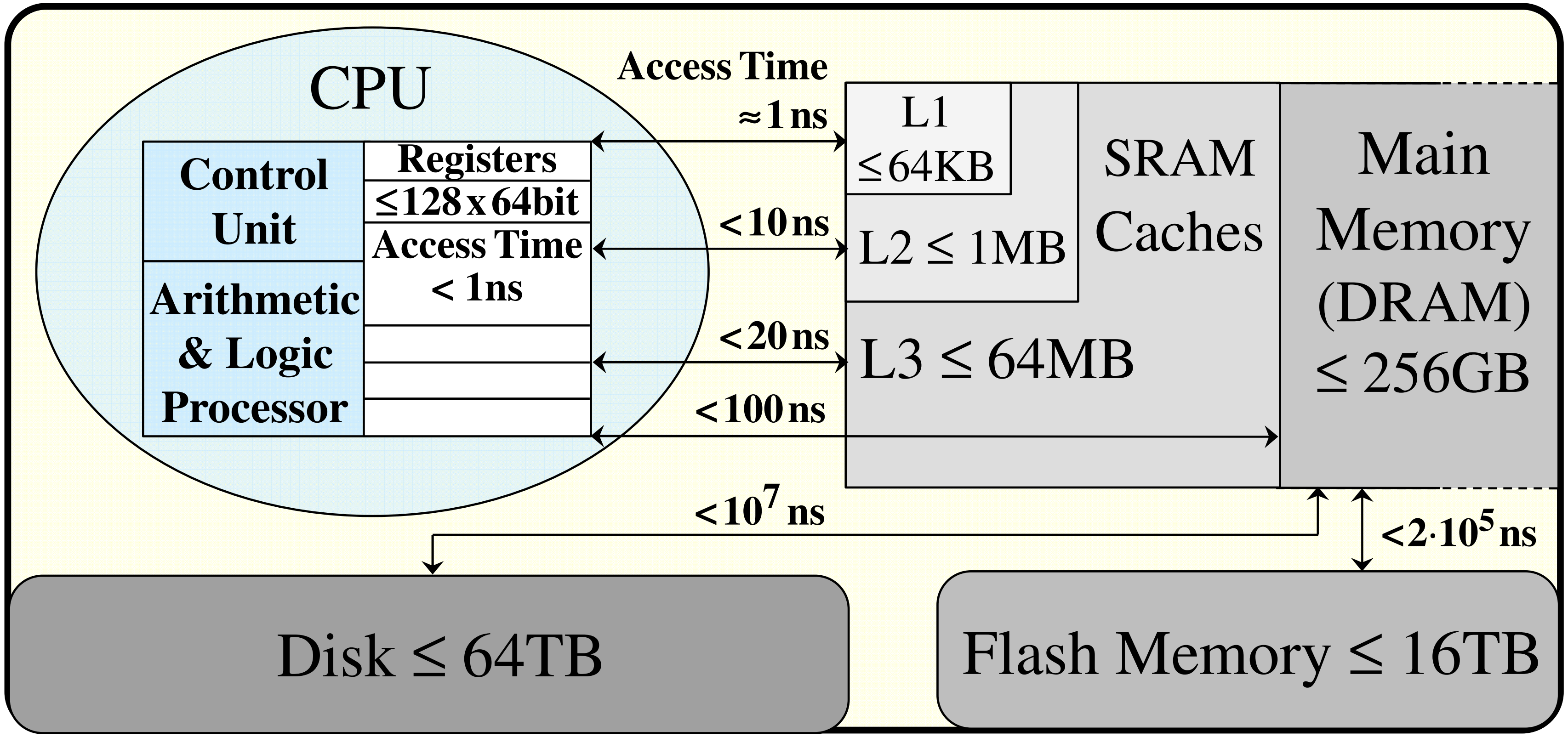
Performance Analysis of Basic Web Caching Strategies (LFU, LRU, FIFO, etc.) with Time-to-live Data Validation

G. Hasslinger, K. Ntougias, F. Hasslinger, O. Hohlfeld

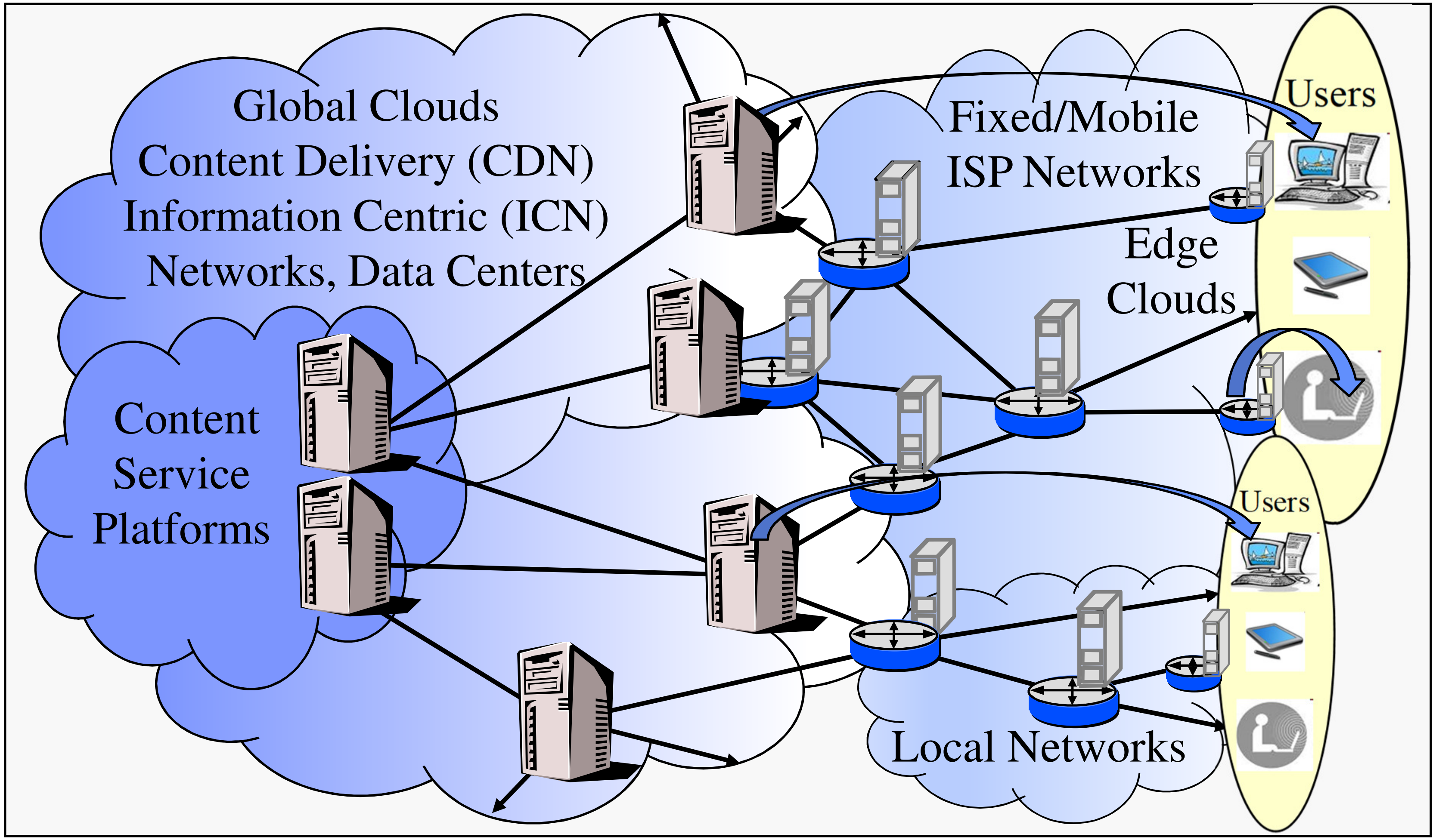
- Caching everywhere in IP networks, on servers etc.
- TTL data validation combined with caching strategies
- Analysis of LFU, LRU, FIFO cache hit ratio with TTL limits
- Evaluations including improved strategy and hit ratio bound
- Extensions & Conclusions

Cache Storage Hierarchy in Web Servers and CPU Systems



Figures for servers from: J.L. Hennessy and D.A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann (2019)

Caching on the Internet for Reduced Delay & Traffic Load



Web Caching Strategies and Data Consistency

- Update Strategies in Caches of Fixed Size

  LRU/LFU: Evict least recently/frequently used data
  FIFO: First-in-first-out: Fastest update scheme
  Score-based, Greedy Dual & Machine Learning Strategies:
  Improve cache utility based on properties per data object

- TTL (Time-to-live) caches:

  Data in the cache is valid for limited time:
  Pure TTL caches vary in size for currently valid data
  Used for DNS caches etc.: Small data units with high churn

- Caches of fixed size also need TTL validation control

Cache Data Consistency: Demands and Practice

- Telemedien Gesetz § 9 (German Law):

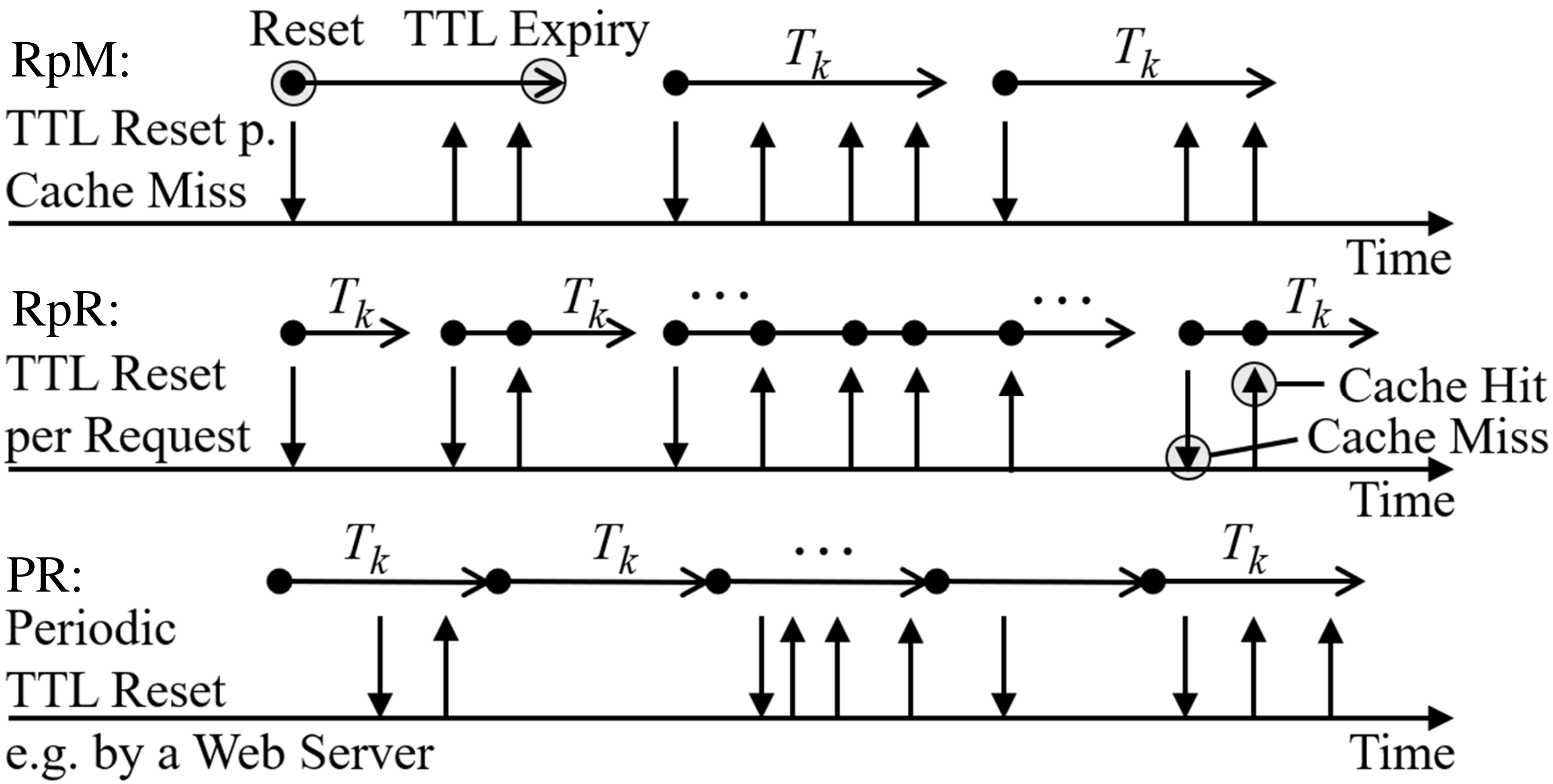
  Allows caching for accelerated or more efficient data transfer, where data is stored for a limited time and, if approved industry standards for data updates are met (RFC 9111: HTTP Caching)

- Zheng et al. (2022): Analysis of LRU with cache invalidation

  "Most web applications apply validation rather than invalidation to maintain cache consistency due to the extra overhead on the network caused by the latter"

- Analysis & evaluation of caches of fixed size with time-to-live (TTL) validation seems to be new !?

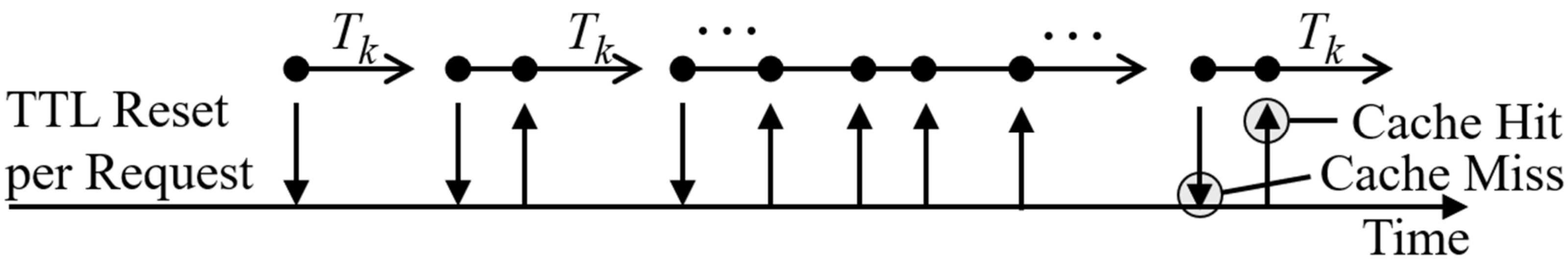
TTL Reset Variants



PR: Periodic resets: Is useful for data validation under web server control

RpM: Reset per miss: Is useful for data validation controlled by caches

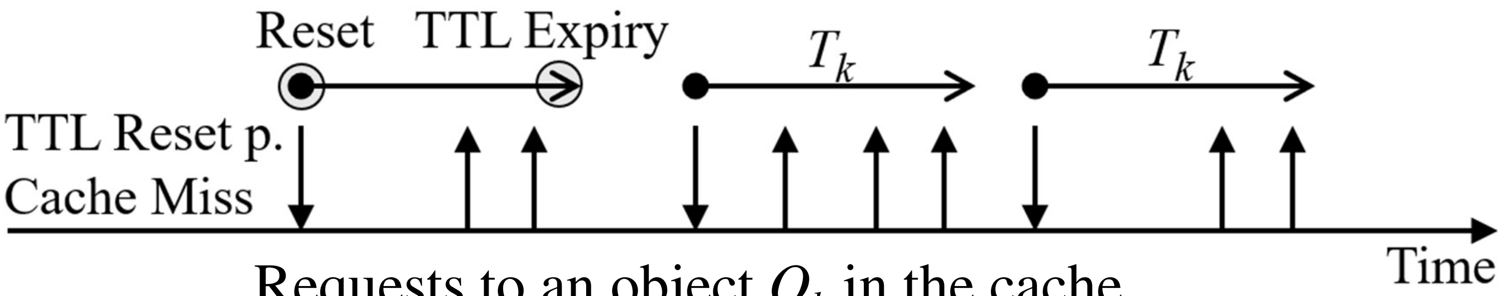
Result for Static Caches of Fixed Size with TTL $R_k = \lambda T_k$ and Reset per Request (RpR)



Requests to an object O_k in the cache
(λ: Entire request arrival rate; IRM: Independent Reference Model)

$$h_{Static,IRM}^{TTL,RpR} = \sum_{k:\, O_k \in C} p_k \left(1 - (1 - p_k)^{R_k}\right)$$

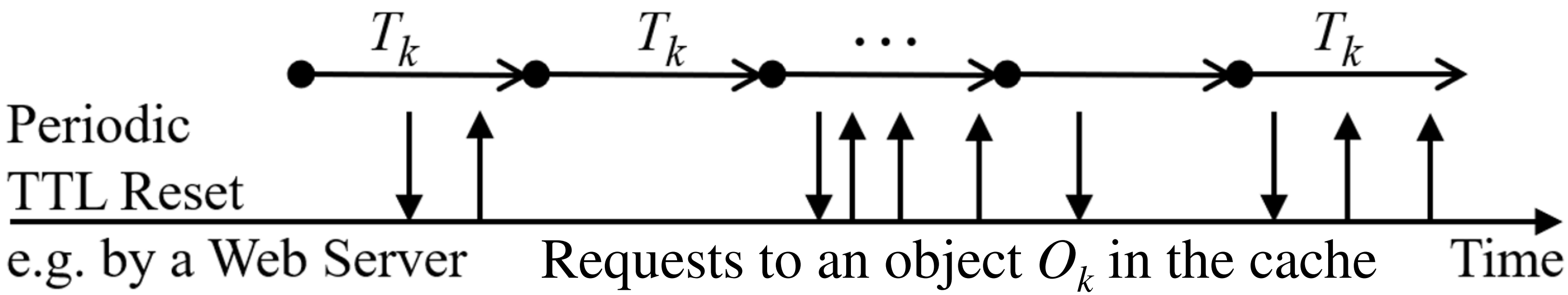
Result for Static Caches of Fixed Size with TTL $R_k = \lambda T_k$ and Reset per Miss (RpM)



Requests to an object O_k in the cache
(λ: Entire request arrival rate)

$$h^{TTL,RpM}_{Static,IRM} = \sum_{k:\, O_k \in C} p_k^2 R_k / (p_k R_k + 1)$$

Result for Static Caches of Fixed Size with TTL $R_k = \lambda T_k$ and Periodic Reset (PR)



(λ: Entire request arrival rate)

$$h^{TTL,PR}_{Static,IRM} = \sum_{k:\,O_k \in C} p_k - \frac{1-(1-p_k)^{R_k+1}}{R_k+1}$$

$$h^{TTL,PR}_{Static} = \sum_{k:\,O_k \in C} \frac{R_k^* - n_k(1-p_k^0)}{R^*} = \sum_{k:\,O_k \in C} p_k^* - \frac{1-p_k^0}{R_k+1}$$

Result holds for arbitrary request pattern based on 2 parameters:
p_k*: % of requests to the object O_k; p_k^0: % of reset intervals with no request

Results for Static Caches of Fixed Size with TTL

$$h^{TTL,RpM}_{Static,IRM} = \sum_{k:\,O_k \in C} p_k^2 R_k / (p_k R_k + 1); \quad (1)$$

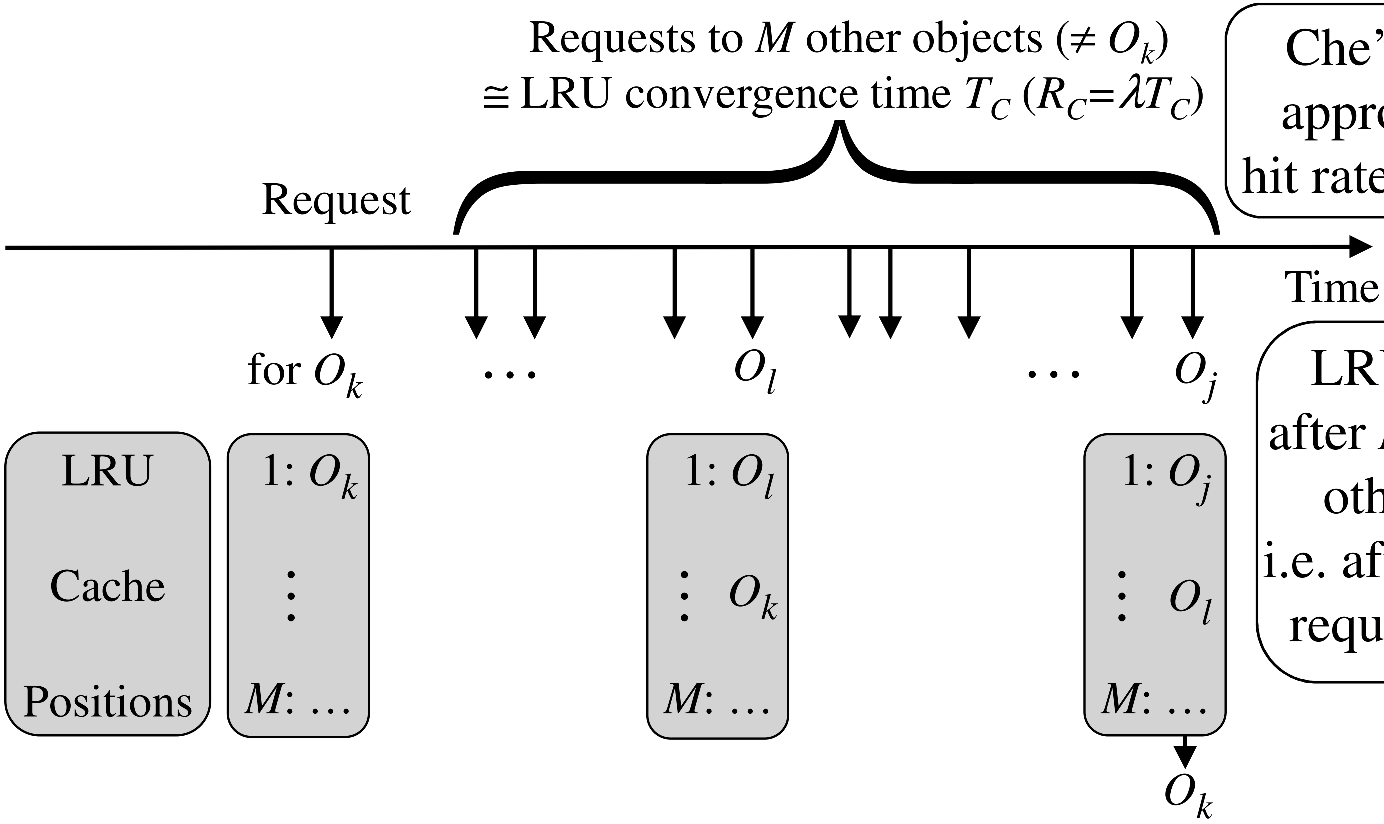
$$h^{TTL,RpR}_{Static,IRM} = \sum_{k:\,O_k \in C} p_k (1 - (1 - p_k)^{R_k}); \quad (2)$$

$$h^{TTL,PR}_{Static,IRM} = \sum_{k:\,O_k \in C} p_k - \frac{1 - (1 - p_k)^{R_k + 1}}{R_k + 1}. \quad (3)$$

$$h^{TTL,PR}_{Static} = \sum_{k:\,O_k \in C} \frac{R_k^* - n_k (1 - p_k^0)}{R^*} = \sum_{k:\,O_k \in C} p_k^* - \frac{1 - p_k^0}{R_k + 1}. \quad (4)$$

RpM: Reset per Miss; IRM: Independent Reference Model
RpR: Reset per Request PM: Periodic Reset

LRU Analysis with TTL and RpR



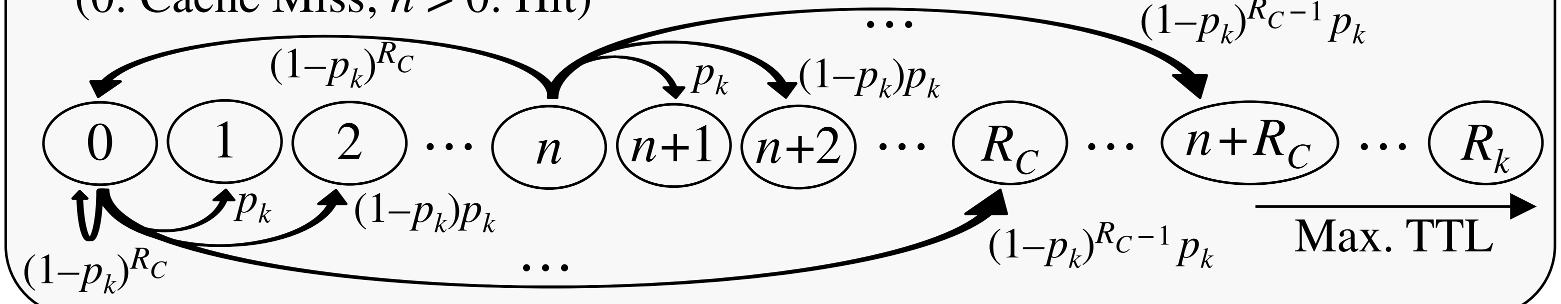
Che's & Fagin's approx. for LRU hit rate are extended

LRU evicts O_k after M requests to other objects, i.e. after time T_C or request count R_C

If the TTL timer T_k of O_k is longer than T_C then O_k is evicted before expiry
LRU with TTL $T_k = T$ and RpR sorts objects according to remaining TTL
$\Rightarrow$ Then the LRU cache is filled with valid objects
$\Rightarrow$ LRU is not affected by TTL for $T > T_C$ and is optimal for $T \leq T_C$

LRU Analysis Results with TTL $R_k = \lambda T_k$

IRM Markov Model for Hits on O_k in an LRU Cache with TTL and RpM
(0: Cache Miss; $n > 0$: Hit)



Reset per Miss:

$$h_k = q_1 + \cdots + q_{R_k}; \quad h^{TTL,RpM}_{LRU,IRM} = \sum_{k=1}^{N} p_k\, h_k$$

q_n: Steady state probability of a hit in state n in the Markov model

Reset per Request:

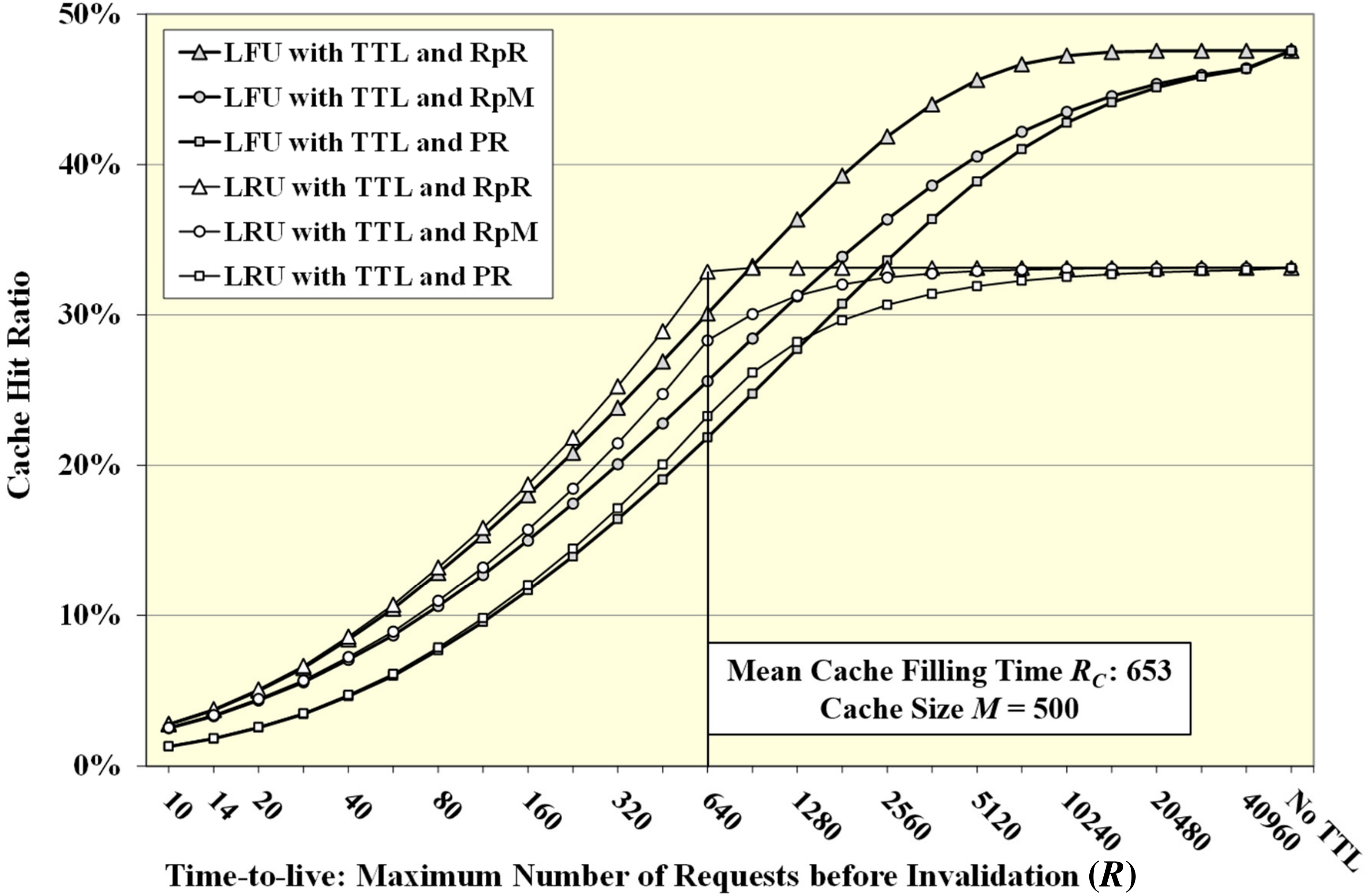
$$h^{TTL,RpR}_{LRU,IRM} = \sum_{k=1}^{N} p_k\left(1 - (1 - p_k)^{\min(R_C, R_k)}\right)$$

Periodic Reset:

$$h_k = \sum_{n=0}^{R_k} \frac{h_k(n)}{R_k + 1} = 1 - \frac{1 - (1 - p_k)^{R_k+1}}{p_k(R_k + 1)} \quad \text{if } R_k \le R_C$$

$$h_k = 1 - \frac{1 + [(R_k + 1 - R_C)p_k - 1](1 - p_k)^{R_C}}{p_k(R_k + 1)} \quad \text{if } R_k \ge R_C$$

IRM hit ratio with unique TTL (& RpR, RpM, PR): LRU is better than LFU for small TTL $R < R_C = \lambda T_C$



FIFO hit ratio analysis with TTL is analogous to LRU

LRU: **Fagin´s or Che´s** approximation for the hit ratio and T_C, R_C
FIFO: **Dan & Towsley´s** approximation for the hit ratio and $T_{C,FIFO}$, R_C

For unique TTL $R = R_k$ per object:
LRU cache is sorted due to remaining valid TTL time for **RpR**
⇒ **LRU** analysis result for reset per **request**:

$$h^{TTL,RpR}_{LRU,IRM} = \sum_{k=1}^{N} p_k (1 - (1 - p_k)^{\min(R_C, R_k)})$$

where $R_C = \lambda T_C$ is the request limit until expiry

FIFO cache is sorted due to remain. valid TTL time for **RpM if** $R_k > R_C$
⇒ **FIFO** analysis result for reset per **miss**:

$$h^{TTL,RpM}_{FIFO,IRM} = \sum_{k=1}^{N} p_k \min(R_C, R_k) / [\min(R_C, R_k) + 1/(\lambda p_k)]$$

where $R_C = \lambda T_{C,FIFO}$ is the request limit until expiry;

Upper IRM hit ratio bound with TTL

- Static caching of most popular objects (LFU)
  $\Rightarrow$ Max. IRM hit ratio without TTL

- Optimum IRM strategy with TTL and RpR, RpM, PR is dynamic:
  Put most popular data into cache, which is currently not expired!

- Optimum IRM caching with TTL may need to upload data, which is currently not requested $\Rightarrow$ no usual caching strategy (!) but can be analyzed:

- Key for analysis: $p_k^{\#Valid}(l)$: Prob. that *k* of the top-*l* objects are valid

  Iterative comp.: $p_k^{\#Valid}(l) = p_{k-1}^{\#Valid}(l)(1-p_k^{Valid})+p_{k-1}^{\#Valid}(l-1)p_k^{Valid}$

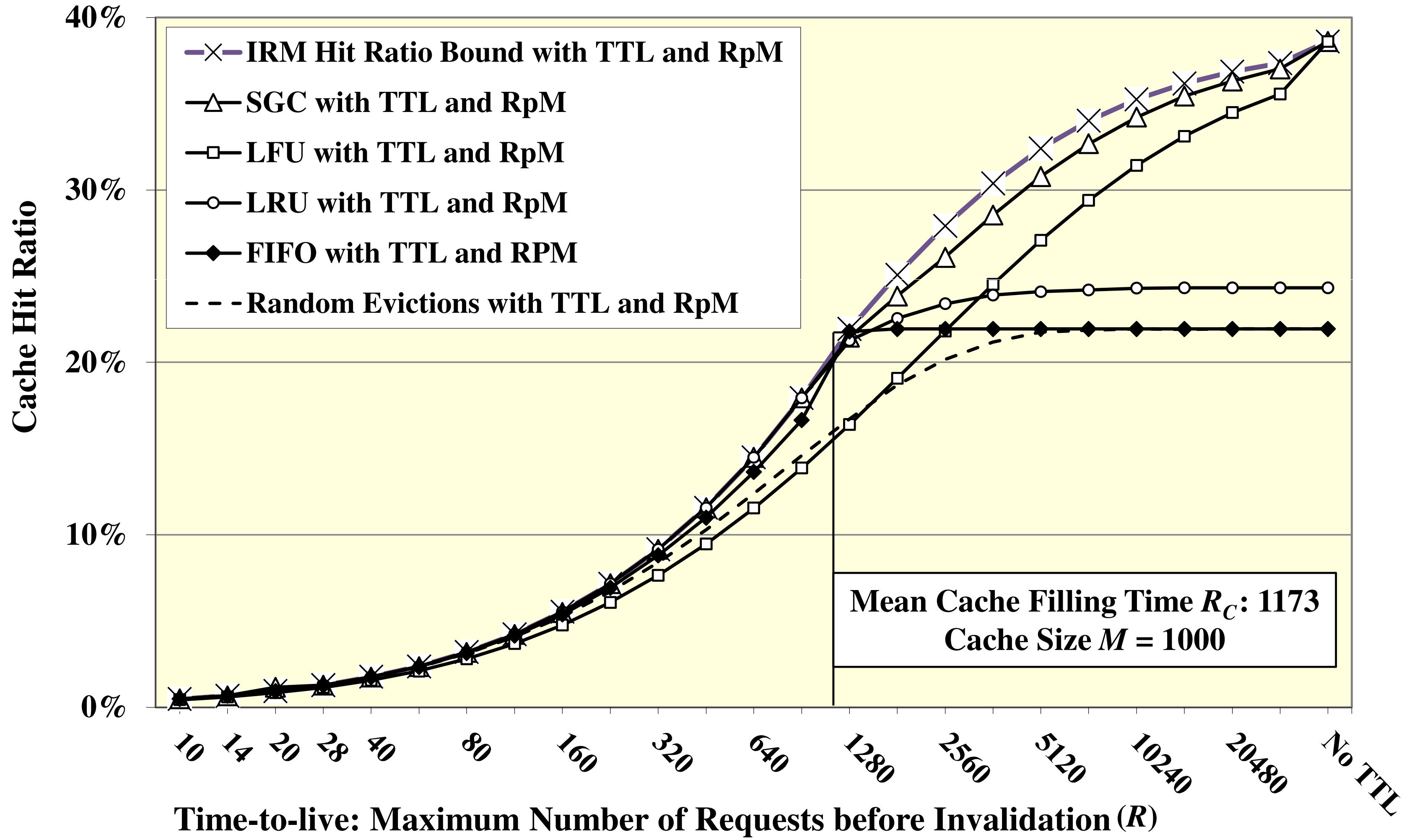
  and $p_k^{Valid} = p_k R_k / (p_k R_k + 1)$ for RpM

- $\Rightarrow$ Prob. that the top-*l* object is cached for obtaining max. hit ratio

A Score-Gated Clock (SGC) strategy for improved hit ratio regarding popularity and TTL counter per object

- SGC caches the top-*M* objects (like LFU) if they are valid
- Other requested objects can replace an (almost) expired top-*M* or another less popular object in the cache
- Eviction candidates are found by a clock scan through the cache with a clock step per request; Clock is the fastest update scheme
- An evicted top-*M* object is restored upon request
- In this way, SGC improves LRU, LFU etc. on the entire TTL range; SGC performance is close to the optimum IRM hit ratio

IRM hit ratio evaluation with TTL and reset per miss: SGC is aware of popularity & TTL to approach bound



Conclusions & Extensions

- IRM Analysis of LFU, LRU, FIFO cache hit ratio with TTL limits leads to explicit results, Markov models, or as extensions of approved approximations for LRU & FIFO
- An analytical hit bound is included and a close to optimum SGC method regarding popularity & current TTL per object
- Further extensions may be worked out:
  - for the analysis of the random eviction principle,
  - for more evaluations with different TTLs per object,
  - for caching of data of different size and value,
  - for correlated request pattern beyond IRM and,
  - for optimized score-based methods for all reset variants.

References

[1] P. Cao and C. Liu, “Maintaining strong cache consistency in the world wide web,” IEEE Trans. Computers, vol. 47(4), pp. 445–457, 1998.
[2] E. Cohen, E. Halperin, and H. Kaplan, “Performance aspects of distributed caches using TTL-based consistency,” Proc. 28th ICALP, Crete, Greece, pp. 744–756, 2001.
[3] Bundesministerium der Justiz, “Telemediengesetz (TMG),” https://www.gesetze-im-internet.de/tmg/ 9.html, 2007.
[4] R. Fielding, M. Nottingham, and J. Reschke, “HTTP caching,” IETF Internet standards track, RFC 9111, 2022.
[5] Q. Zheng et al., “On the analysis of cache invalidation with LRU replacement,” IEEE Trans. TPDS, vol. 33/3, pp. 654–666, 2022.
[6] J. Jung, A. Berger, and H. Balakrishnan, “Modelling TTL-based Internet caches,” Proc. IEEE Infocom, pp. 417–426, 2003.
[7] J. Shim, P. Scheuermann, and R. Vingralek, “Proxy cache algorithms: Design, implementation, performance,” IEEE Trans. Know. Data Engin., 11(4), pp. 549–562, 1999.
[8] J. Yang, Y. Yue, and K. Rashmi, “A large-scale analysis of key-value cache clusters at Twitter,” ACM Trans. Storage, vol. 17(3)17, 2021.
[9] D. S. Berger, P. Gland, S. Singla, and F. Ciucu, “Exact analysis of TTL cache networks,” Perform. Eval., vol. 79, pp. 2–23, 2014.
[10] M. Dehghan et al., “A utility optimization approach to network cache design,” IEEE/ACM Trans. Networking, pp. 1013–1027, 2019.
[11] G. Hasslinger et al., “An overview of analysis methods and evaluation results for caching strategies,” Computer Networks, vol. 228, 2023.
[12] M. Garetto, E. Leonardi, and V. Martina, “A unified approach to the performance analysis of caching systems,” TOMPECS, vol. 1(3)12, pp. 1–28, 2016.
[13] G. Hasslinger et al., “Performance evaluation for new web caching strategies combining LRU with score based object selection,” COMNET 125, pp. 172–186, 2017.
[14] W. King III, “Analysis of paging algorithms,” Proc. IFIP Congress, Ljublanjana, Yugoslavia, pp. 485–490, 1971.
[15] R. Fagin, “Asymptotic miss ratios over independent references,” J. Comput. Syst. Sci., vol. 14, no. 2, pp. 222–250, 1977.
[16] H. Che, Y. Tung, and Z. Wang, “Hierarchical web caching systems: modeling, design and experimental results,” IEEE JSAC, vol. 20(7), pp. 1305–1314, 2002.
[17] G. Hasslinger et al., “Scope and accuracy of analytic and approximate results for FIFO, clock-based and LRU caching performance,” Future Internet 15/3, pp. 1–17, 2023.
[18] A. Dan and D. F. Towsley, “An approximate analysis of the LRU and FIFO buffer schemes,” Proc. ACM SIGMETRICS, Boulder, Colorado, USA, pp. 143–152, 1990.
[19] C. Fricker, P. Robert, and J. Roberts, “A versatile and accurate approximation for LRU cache performance,” 24th ITC Congress, Krakow, Poland, pp. 1–8, 2012.