

# On Models for Performance Analysis of a Core Cache Network and Power Save of a Wireless Access Network

Ph.D. Thesis defended by

Nicaise E. CHOUNGMO FOFACK

Advisors: Philippe Nain and Sara Alouf



# Outline

Introduction

Single cache

Cache networks under i.i.d. requests

Conclusion

# Outline

## Introduction

- Context

- In-network caching

- Single cache

- Cache networks under i.i.d. requests

- Conclusion

# Motivations

## Focus

- 1 Power save protocols
  - Reduce energy consumption

# Motivations

## Focus

- 1 Power save protocols
  - Reduce energy consumption
- 2 In-network caching
  - Reduce network traffic
  - Improve users QoE

# Problem 1: Power save protocols

## Energy consumption problem (Statement)

- 1 Study case: 3/4G cellular networks (LTE, WiMAX)
- 2 Issue: Continuous connectivity  $\implies$  Increase energy costs on TRX
- 3 Challenges:
  - Discontinuous TRX protocols by 3GPP
  - Realistic web traffic (non-Poisson)
  - Both mobiles (UEs) and Node B (eNB)

# Problem 1: Power save protocols

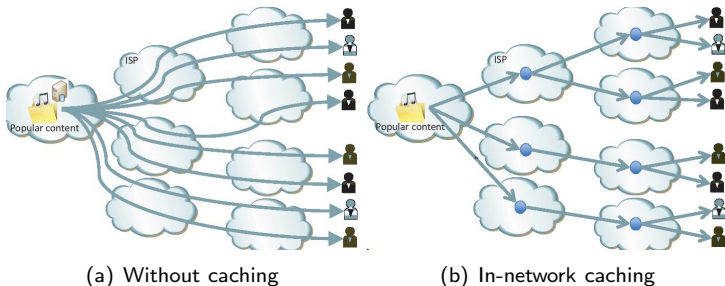
## Energy consumption problem (models)

- 1 Performance metrics: Gains (UE & eNB), Delay, Power Ratio, Download time
- 2 Outcomes:
  - Optimal configuration of Discontinuous TRX protocols
  - Impact of inputs on the metrics of interest
  - Lessons: Up to 90% TRX costs, #UEs < 350, small web page, & small # embedded objects
- 3 S. Alouf, V. Mancuso, N. C. F., "Analysis of power saving and its impact on web traffic in cellular networks with continuous connectivity", *Pervasive and Mobile Computing*, 8(5):646-661, Oct. 2012

## Problem 2: “On-demand” cache networks

### Benefits

- 1 Reduce network traffic
- 2 Improve Users QoE





## Problem 2: “On-demand” cache networks

### Examples

- Domain Name System  
(worldwide, everyone uses it)
- CDNs and VoD Systems  
(Akamai, Google, Youtube, Orange)
- Content-Centric Network  
(CCN, Xerox Parc, Jacobson et al.)

## Problem 2: “On-demand” cache networks

### Issues

- 1 Design large scale cache networks
- 2 Models for performance evaluation cache nets
- 3 Optimization and control of “on-demand” caches (QoS, DiffServ)

## Problem 2: “On-demand” cache networks

### Challenges & limitations of existing models

- ❶ Classical policies (LRU, FIFO, RND, etc.) not flexible enough
- ❷ No exact results even for Tandem of 2 caches
- ❸ Only approximations, limited by
  - Traffic models, IRM or Poisson processes
  - Topology and Routing, Tree-centric
  - Heterogeneity, Same policy everywhere
  - Accuracy and Complexity of performance models

# Outline

Introduction

Single cache

- Cache architecture

- Practical concerns

Cache networks under i.i.d. requests

Conclusion

# Time-To-Live (TTL)-based policy

## How does it work?

- ▶ On cache miss: add content, assign timer  $T$  to it
- ▶ On cache hit: two possibilities
  - ▶ Re-initialize timer  $T$
  - ▶ Use remaining value of  $T$
- ▶ Remove content **(only) when** timer  $T$  expired

# Time-To-Live (TTL)-based policy

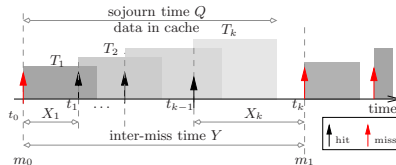
## How does it work?

- ▶ On cache miss: add content, assign timer  $T$  to it
- ▶ On cache hit: two possibilities
  - ▶ Re-initialize timer  $T$
  - ▶ Use remaining value of  $T$
- ▶ Remove content **(only) when** timer  $T$  expired

# Time-To-Live (TTL)-based policy

## How does it work?

- ▶ On cache miss: add content, assign timer  $T$  to it
- ▶ On cache hit: two possibilities
  - ▶ Re-initialize timer  $T$
  - ▶ Use remaining value of  $T$
- ▶ Remove content (**only**) when timer  $T$  expired

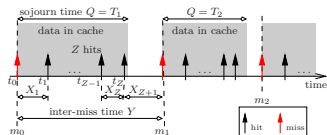


(c) TTL-renew,  $r = 1$

# Time-To-Live (TTL)-based policy

## How does it work?

- ▶ On cache miss: add content, assign timer  $T$  to it
- ▶ On cache hit: two possibilities
  - ▶ Re-initialize timer  $T$
  - ▶ Use remaining value of  $T$
- ▶ Remove content (**only**) when timer  $T$  expired



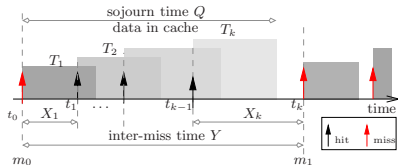
(d) TTL-non-renew,  $r = 0$



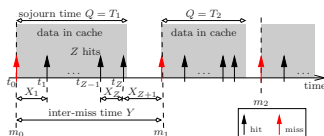
# Time-To-Live (TTL)-based policy

## How does it work?

- ▶ On cache miss: add content, assign timer  $T$  to it
- ▶ On cache hit: two possibilities
  - ▶ Re-initialize timer  $T$
  - ▶ Use remaining value of  $T$
- ▶ Remove content **(only) when** timer  $T$  expired



(e) TTL-renew,  $r = 1$



(f) TTL-non-renew,  $r = 0$

# Time-To-Live (TTL)-based policy

## How does it work?

- ▶ On cache miss: add content, assign timer  $T$  to it
- ▶ On cache hit: two possibilities
  - ▶ Re-initialize timer  $T$
  - ▶ Use remaining value of  $T$
- ▶ Remove content **(only) when** timer  $T$  expired

	LRU	RND	FIFO	TTL
Space	finite	finite	finite	Infinite
Time	unknown	unknown	unknown	Limited
Control	no way	no way	no way	yes, timers (rvs)

# Time-To-Live (TTL)-based policy

## How does it work?

- ▶ On cache miss: add content, assign timer  $T$  to it
- ▶ On cache hit: two possibilities
  - ▶ Re-initialize timer  $T$
  - ▶ Use remaining value of  $T$
- ▶ Remove content **(only) when** timer  $T$  expired

## TTL-based models are also interesting...

- ▶ extend easily to networks
- ▶ use to accurately describe LRU, RND, FIFO policies

# Model

## Assumptions

1. Cache size is **INFINITE** and **ONLY TTLs** are used
2. Delays are negligible

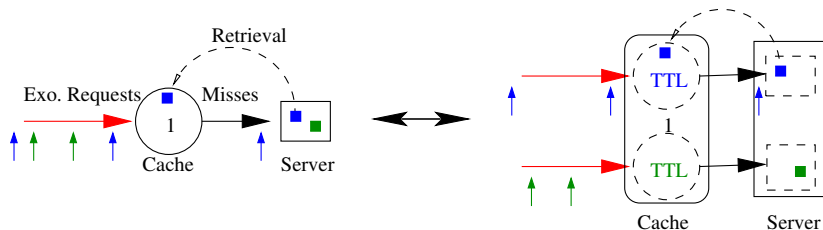


Figure: TTL decoupling effect  $\Rightarrow$  Focus on **SINGLE CONTENT!**

# Model

## Assumptions

1. Cache size is **INFINITE** and **ONLY TTLs** are used
2. Delays are negligible

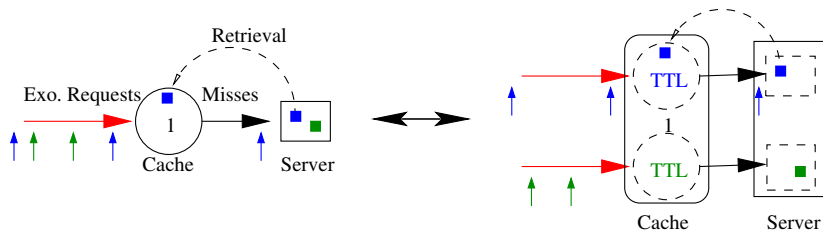
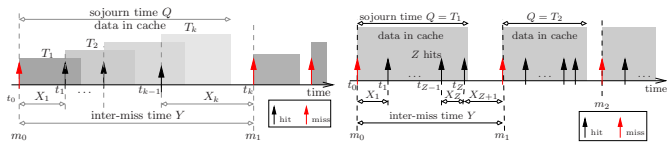


Figure: TTL decoupling effect  $\Rightarrow$  Focus on **SINGLE CONTENT!**

# Hit and Occupancy probs

- File requests form a **stationary point process**,  $\{\mathcal{N}(t), t \geq 0\}$

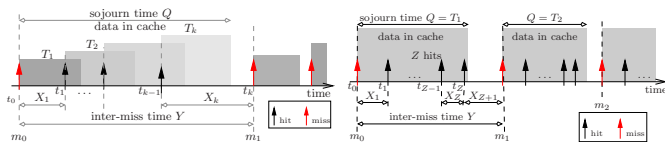


(a) Case  $r = 1$ , Renew TTL

(b) Case  $r = 0$ , Non-renew TTL

# Hit and Occupancy probs

- File requests form a **stationary point process**,  $\{\mathcal{N}(t), t \geq 0\}$

(c) Case  $r = 1$ , Renew TTL(d) Case  $r = 0$ , Non-renew TTL

## Theorem (Stationary case)

$$H_P = r \mathbb{P}^0(X_1 < T_1) + \bar{r} \left( 1 - \left( 1 + \mathbb{E}^0[\mathcal{N}(T_1)] \right)^{-1} \right) \quad (1)$$

$$O_P = \left( \mathbb{E}^0[X_1]^{-1} \times (1 - H_P) \right) \times \mathbb{E}^0[Q], \quad \text{“Mean-Value Formula”} \quad (2)$$

$Q$  is the total sojourn time given by:

$$Q = \begin{cases} T_1 & \text{if } r = 0 \\ T_1 \times \mathbf{1}\{X_1 > T_1\} + (X_1 + \tilde{Q}) \times \mathbf{1}\{X_1 < T_1\} & \text{if } r = 1. \end{cases} \quad (3)$$

# Miss process

- ▶ Arrivals occur at jumps of a DTMC  $\{\xi_i, i \geq 0\}$  on  $\mathcal{S} = \{1, 2, \dots, J\}$ ,
- ▶ Requests form a **stationary Markov Renewal Process (MRP)**,  
 $[\mathbf{F}(t)]_{j,k} := \mathbb{P}(X_i < t, \xi_{i+1} = j \mid \xi_i = k)$ ,  $T(t) = \mathbb{P}(T_1 < t)$



## Miss process

- ▶ Arrivals occur at jumps of a DTMC  $\{\xi_i, i \geq 0\}$  on  $\mathcal{S} = \{1, 2, \dots, J\}$ ,
- ▶ Requests form a **stationary Markov Renewal Process (MRP)**,  
 $[\mathbf{F}(t)]_{j,k} := \mathbb{P}(X_i < t, \xi_{i+1} = j \mid \xi_i = k)$ ,  $T(t) = \mathbb{P}(T_1 < t)$

### Proposition (Kernel of the miss process)

Miss process is a **MRP**, with kernel

$$\mathbf{G}^{(0)}(t) = \mathbf{F}(t) - \int_0^t d\mathbf{R}(x)(\mathbf{I} - \mathbf{F}(t-x)) \quad (4)$$

$$\mathbf{G}^{(1)}(t) = \mathbf{F}(t) - \mathbf{L}(t) + \int_0^t d\mathbf{L}(x)\mathbf{G}^{(1)}(t-x) \quad (5)$$

$$[\mathbf{L}(t)]_{j,k} := \int_0^t (1 - T(x)) dF_{j,k}(x), \quad [\mathbf{R}(t)]_{j,k} := \int_0^t (1 - T(x)) dM_{j,k}(x)$$

## Miss process

- ▶  $\mathcal{S} = \{1\} \implies$  **Renewal process**,  $F(t) = \mathbb{P}(X_1 < t)$ ,  $M(t) = \mathbb{E}[\mathcal{N}(t)]$

### Corollary (CDF of the inter-miss times)

*Miss process is a renewal process, and  $\mathbb{P}(Y < t)$*

$$G^{(0)}(t) = F(t) - \int_0^t (1 - T(x)) dM(x) (1 - F(t - x))$$

$$G^{(1)}(t) = F(t) - \int_0^t (1 - T(x)) dF(x) + \int_0^t (1 - T(x)) dF(x) G^{(1)}(t - x)$$

Hint,  $\mathcal{S} = \{1\}$ .

## Miss process

- ▶  $\mathcal{S} = \{1\} \implies$  **Renewal process**,  $F(t) = \mathbb{P}(X_1 < t)$ ,  $M(t) = \mathbb{E}[\mathcal{N}(t)]$

### Corollary (CDF of the inter-miss times)

*Miss process is a renewal process, and  $\mathbb{P}(Y < t)$*

$$G^{(0)}(t) = F(t) - \int_0^t (1 - T(x)) dM(x) (1 - F(t - x))$$

$$G^{(1)}(t) = F(t) - \int_0^t (1 - T(x)) dF(x) + \int_0^t (1 - T(x)) dF(x) G^{(1)}(t - x)$$

Hint,  $\mathcal{S} = \{1\}$ .

$$Y = \begin{cases} X_1 + \dots + X_{\mathcal{N}(T_1)+1} & \text{if } r = 0 \\ X_1 \mathbf{1}(X_1 > T_1) + (X_1 + \tilde{Y}) \mathbf{1}(X_1 < T_1) & \text{if } r = 1. \end{cases}$$

# How to set up a TTL-based model?

## Proposition (Optimal TTL distribution)

If  $F(t)$  is *concave* then *Deterministic TTL* performs the best.

# How to set up a TTL-based model?

## Proposition (Optimal TTL distribution)

If  $F(t)$  is *concave* then *Deterministic TTL* performs the best.

### Hint.

- ▶ Consider  $T$  and  $D$  s.t.  $\mathbb{E}[T] = D$
- ▶ show  $D \leq_{\text{CX}} T$  (Jensen's Inequality)
- ▶ Express  $H_P = \mathbb{E}[f(T)]$  and  $O_P = \mathbb{E}[g(T)]$ ,  $f$  concave,  $g$  cvx

# How to set up a TTL-based model?

## Proposition (Optimal TTL distribution)

If  $F(t)$  is *concave* then *Deterministic TTL* performs the best.

### Hint.

- ▶ Consider  $T$  and  $D$  s.t.  $\mathbb{E}[T] = D$
- ▶ show  $D \leq_{\text{CX}} T$  (Jensen's Inequality)
- ▶ Express  $H_P = \mathbb{E}[f(T)]$  and  $O_P = \mathbb{E}[g(T)]$ ,  $f$  concave,  $g$  cvx

## Remark (Conjecture... from experiments)

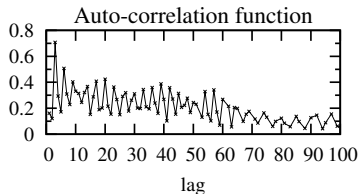
Otherwise, TTL distr. with *strictly positive*  $c_V^2$  (e.g. hypo/exp/hyper).

# Validation with real data: traces from Inria DNS cache

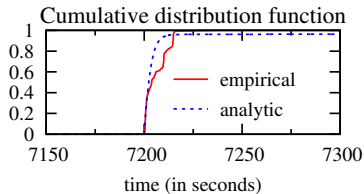
- ▶ TTL-non-renewing cache, 6-th most popular content, TTL  $T = 2h$
- ▶ Match higher order and joint moments using Markov Arrival Process
- ▶ Use analytic CDF  $F(t)$  as input of TTL model

# Validation with real data: traces from Inria DNS cache

- ▶ TTL-non-renewing cache, 6-th most popular content, TTL  $T = 2h$
  - ▶ Match higher order and joint moments using Markov Arrival Process
  - ▶ Use analytic CDF  $F(t)$  as input of TTL model
- ▶ Inter-requests are correlated & good approximation of  $G(t)$



(g) ACF inter-arrival times



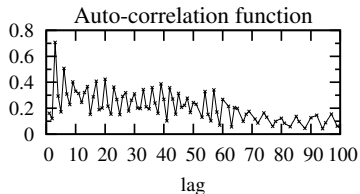
(h) CDF inter-miss times



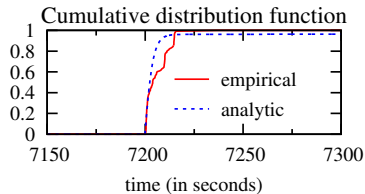
# Validation with real data: traces from Inria DNS cache

- ▶ **TTL-non-renewing** cache, 6-th most popular content, TTL  $T = 2h$
- ▶ Match higher order and joint moments using Markov Arrival Process
- ▶ Use analytic CDF  $F(t)$  as input of TTL model

- ▶ **Inter-requests are correlated & good approximation of  $G(t)$**



(i) ACF inter-arrival times



(j) CDF inter-miss times

- ▶ Performance metrics: **accurate predictions**

Metric	Trace	Model	Rel. err. (%)
miss rate	0.00013876	0.00013749	0.920
hit prob.	0.99943	0.99941	0.002
occupancy	0.99914	0.98995	0.920

# Accounting for constraints: Practical (Pra)-TTL

## Pra-TTL: Deployment of TTL model

- ▶ Cache miss: add content and assign TTL
- ▶ Timer expiration: keep it in cache
- ▶ Buffer full: drop content that expired (or will expire sooner)

How to calculate TTLs? ...subject to constraints

# Accounting for constraints: Practical (Pra)-TTL

## Pra-TTL: Deployment of TTL model

- ▶ Cache miss: add content and assign TTL
- ▶ Timer expiration: keep it in cache
- ▶ Buffer full: drop content that expired (or will expire sooner)

## How to calculate TTLs? ...subject to constraints

- ▶ Energy / Holding costs:  $\sum_f (1 - H_{P,f}) c_{t,f} + O_{P,f} c_{h,f} \leq \delta$
- ▶ Refresh rate:  $M_{R,f} \geq \eta$ , (e.g. DNS cache)
- ▶ Capacity:  $\sum_f O_{P,f} = B$ , (e.g. Web cache)

# TTL model prediction versus Pra-TTL simulation results

- ▶ Linear network of 5 Exp. TTL-based caches
- ▶ Constraint: Cache size  $B = 20$
- ▶ Experiment: Poisson ( $\lambda_1 = 2$ ), Zipf( $\alpha = 1.2$ ), Catalog size  $N = 200$

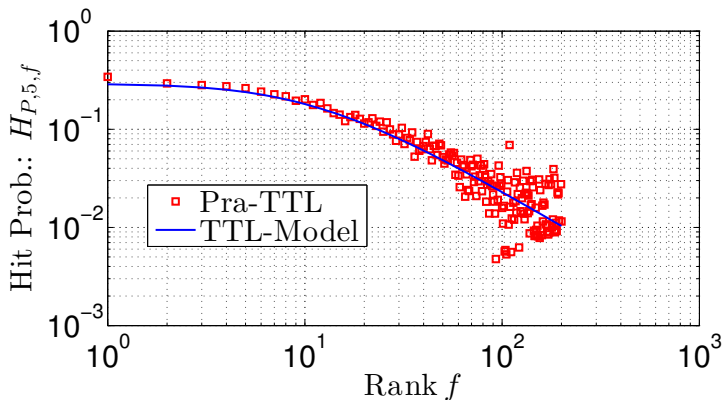
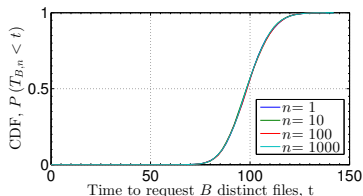


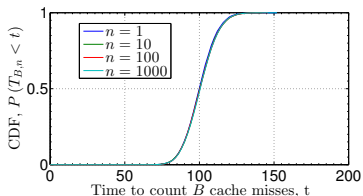
Figure: Cache 5 (root node)

# TTL models and LRU, FIFO, RND

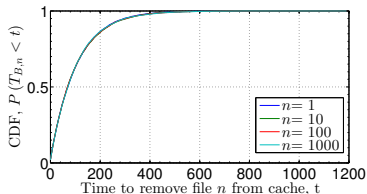
- ▶ Cache of size  $B$ , Content  $n$  requested **ONLY ONCE**, time spent  $T_{B,n}$
- ▶ Experiment: Poisson ( $\lambda = 1$ ), Zipf( $\alpha = 0.65$ ),  $B = 10^2$ , Catalog size  $N = 10^4$



(a) LRU as a TTL-renewing cache



(b) FIFO as TTL-non-renewing cache



(c) RND as TTL-(non)-renewing cache

# Outline

Introduction

Single cache

Cache networks under i.i.d. requests

    Hierarchical networks: Tree/Polytree

    General cache networks

Conclusion

# Step by step analysis (I)

## Multiple sources

Merging renewal request streams, CDF of first inter-request time

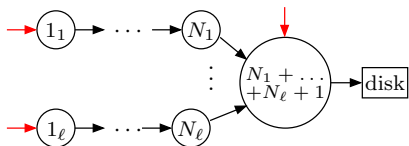


Figure: Exact results on Linear-star networks

# Step by step analysis (I)

## Multiple sources

Merging renewal request streams, CDF of first inter-request time

### 1. Lawrence's Theorem $\implies$ Exact CDF

$$1 - A(t) = \sum_{k=1}^K \frac{\lambda_k}{\sum_{l=1}^K \lambda_l} (1 - A_k(t)) \prod_{j=1, j \neq k}^K \lambda_j \int_t^{\infty} (1 - A_j(u)) du \quad (4)$$



# Step by step analysis (I)

## Multiple sources

Merging renewal request streams, CDF of first inter-request time

### 1. Lawrence's Theorem $\implies$ Exact CDF

$$1 - A(t) = \sum_{k=1}^K \frac{\lambda_k}{\sum_{l=1}^K \lambda_l} (1 - A_k(t)) \prod_{j=1, j \neq k}^K \lambda_j \int_t^{\infty} (1 - A_j(u)) du \quad (4)$$

### 2. Poisson approx. $\implies$ Exponential CDF

$$A(t) = 1 - e^{-\Lambda t}, \quad \Lambda = \sum_{l=1}^K \lambda_l, \quad c_p^2 = 1 \quad (5)$$

# Step by step analysis (I)

## Multiple sources

Merging renewal request streams, CDF of first inter-request time

### 1. Lawrence's Theorem $\implies$ Exact CDF

$$1 - A(t) = \sum_{k=1}^K \frac{\lambda_k}{\sum_{l=1}^K \lambda_l} (1 - A_k(t)) \prod_{j=1, j \neq k}^K \lambda_j \int_t^{\infty} (1 - A_j(u)) du \quad (4)$$

### 2. Poisson approx. $\implies$ Exponential CDF

$$A(t) = 1 - e^{-\Lambda t}, \quad \Lambda = \sum_{l=1}^K \lambda_l, \quad c_p^2 = 1 \quad (5)$$

### 3. Whitt approx. $\implies$ 2-state Hyper-exp. / Shifted-exp. CDFs

- ▶ asymptotic method,  $\Lambda$  and  $c_a^2$ , moments of  $\frac{\mathcal{N}(t)}{t}$
- ▶ stationary interval,  $\Lambda$  and  $c_s^2$ , moments of  $X_1$

# Step by step analysis (I)

## Multiple sources

Merging renewal request streams, CDF of first inter-request time

1. Lawrence's Theorem  $\implies$  Exact CDF

$$1 - A(t) = \sum_{k=1}^K \frac{\lambda_k}{\sum_{l=1}^K \lambda_l} (1 - A_k(t)) \prod_{j=1, j \neq k}^K \lambda_j \int_t^{\infty} (1 - A_j(u)) du \quad (4)$$

2. Poisson approx.  $\implies$  Exponential CDF

$$A(t) = 1 - e^{-\Lambda t}, \quad \Lambda = \sum_{l=1}^K \lambda_l, \quad c_p^2 = 1 \quad (5)$$

3. Whitt approx.  $\implies$  2-state Hyper-exp. / Shifted-exp. CDFs

- ▶ asymptotic method,  $\Lambda$  and  $c_a^2$ , moments of  $\frac{\mathcal{N}(t)}{t}$
- ▶ stationary interval,  $\Lambda$  and  $c_s^2$ , moments of  $X_1$

4. Hybrid Approx.  $\implies$   $\Lambda$  and  $c_h^2 = w c_s^2 + (1 - w) c_p^2$

- ▶ combines Poisson and Whitt (stationary interv) approx.
- ▶ provides empirical expressions of  $w$

# Step by step analysis (II)

## Multiple destinations

Splitting a request stream (e.g. Bernoulli routing with prob.  $p$ )

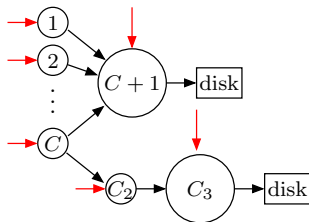


Figure: Polytree network

## Step by step analysis (II)

### Multiple destinations

Splitting a request stream (e.g. Bernoulli routing with prob.  $p$ )

1. Sum of geometric number of i.i.d.  $X_i \implies$  Exact result
2. Poisson, Whitt, and Hybrid approx. (moments matching)

$$E[X_{i,\text{th}}] = \frac{E[X_i]}{p}$$
$$E[X_{i,\text{th}}^2] = \frac{E[X_i^2]}{p} + 2\frac{1-p}{p^2}(E[X_i])^2$$

# Heterogeneous (LRU, RND, or FIFO) tree cache networks

- ▶ 4-ary tree of depth 5 with 341 caches (LRU, RND, or FIFO)
- ▶ Experiment: Poisson requests at leaves ( $\lambda = 1$ ), Zipf( $\alpha = 0.7$ ),  $B \in [50, 150]$ ,  $N = 10^4$

Level 1, Cache 1, Policy:RND, Capacity: 147

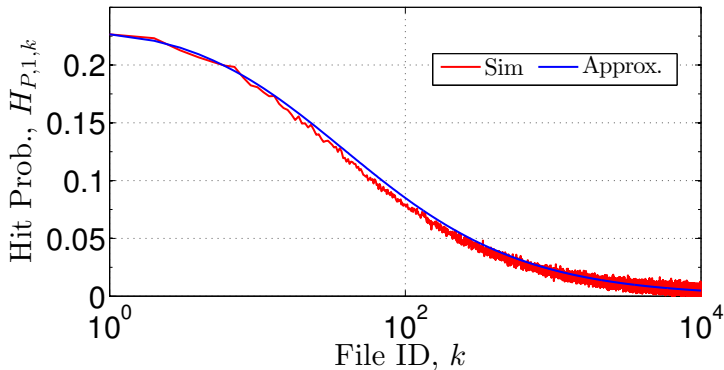


Figure: Hybrid approximation

# Heterogeneous (LRU, RND, or FIFO) tree cache networks

- ▶ 4-ary tree of depth 5 with 341 caches (LRU, RND, or FIFO)
- ▶ Experiment: Poisson requests at leaves ( $\lambda = 1$ ), Zipf( $\alpha = 0.7$ ),  $B \in [50, 150]$ ,  $N = 10^4$

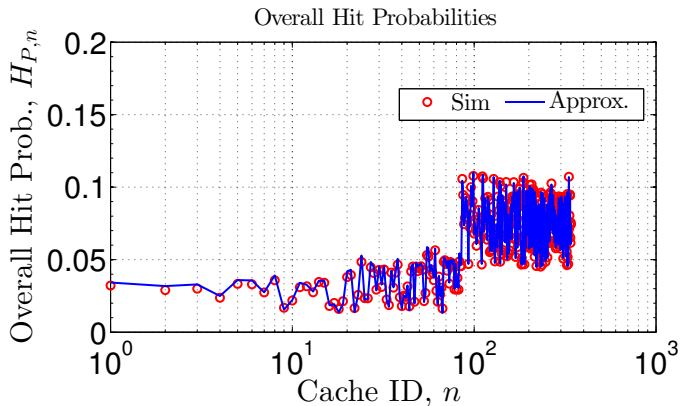


Figure: Hybrid approximation

# Heterogeneous (LRU, RND, or FIFO) tree cache networks

- ▶ 4-ary tree of depth 5 with 341 caches (LRU, RND, or FIFO)
- ▶ Experiment: Poisson requests at leaves ( $\lambda = 1$ ), Zipf( $\alpha = 0.7$ ),  $B \in [50, 150]$ ,  $N = 10^4$

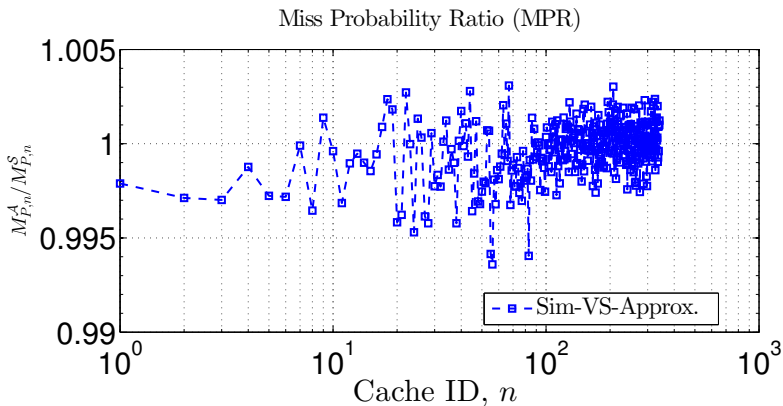
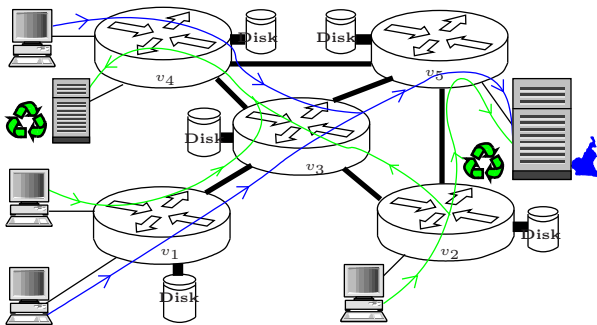


Figure: Hybrid approximation



# Routing topology and State dependencies

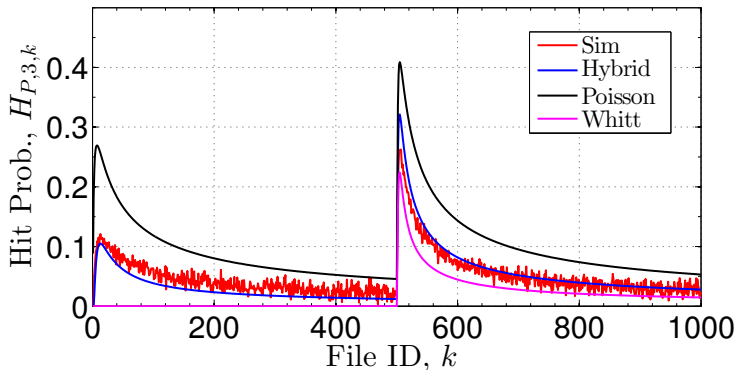
Consider this five caches network



# Iterative procedure for our 5 caches network

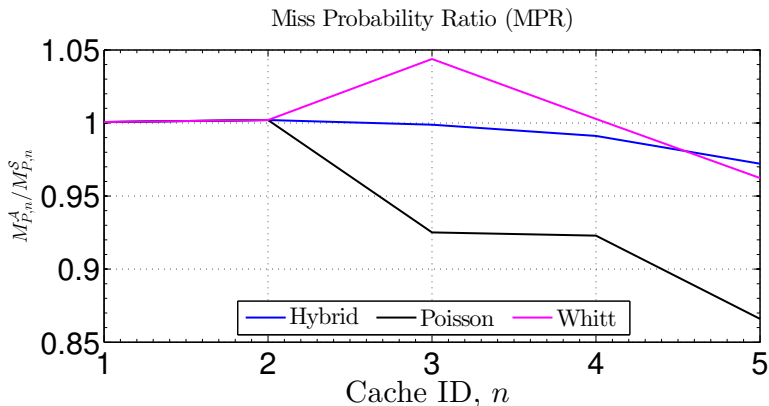
- ▶ LRU policy at all caches, Two classes of contents ( $k = [1..500], [501..1000]$ )
- ▶ Experiment: per-class Poisson requests ( $\lambda = 1$ ), Zipf( $\alpha = 0.7$ ),  $B = 10^2$ ,  $N = 10^3$

Cache 3, Policy:LRU, Capacity: 100



# Iterative procedure for our 5 caches network

- ▶ LRU policy at all caches, Two classes of contents ( $k = [1..500], [501..1000]$ )
- ▶ Experiment: per-class Poisson requests ( $\lambda = 1$ ), Zipf( $\alpha = 0.7$ ),  $B = 10^2$ ,  $N = 10^3$



# Outline

Introduction

Single cache

Cache networks under i.i.d. requests

Conclusion

# Take off message

- 1 Cache networks
  - Think “TTL model”
  - Apply your constraints (capacity, delay, energy, bandwidth,...)
  - Deploy your “TTL model” using LRU, RND, FIFO or Pra-TTL
- 2 New research opportunities
  - Non-zero delays
  - $q$ -LRU and LFU policies
  - Temporal and geographical locality problems
  - QoS and cache optimization problems
- 3 TTL-based behaviors not only in cache networks
  - Economics: Product warranty
  - Physics: Geiger-Müller counters (of particles)
  - Smart device design (idle mode)

# Take off message

- 1 Cache networks
  - Think “TTL model”
  - Apply your constraints (capacity, delay, energy, bandwidth,...)
  - Deploy your “TTL model” using LRU, RND, FIFO or Pra-TTL
- 2 New research opportunities
  - Non-zero delays
  - $q$ -LRU and LFU policies
  - Temporal and geographical locality problems
  - QoS and cache optimization problems
- 3 TTL-based behaviors not only in cache networks
  - Economics: Product warranty
  - Physics: Geiger-Müller counters (of particles)
  - Smart device design (idle mode)

# List of publications

- [1] S. Alouf, V. Mancuso, **N. Choungmo Fofack**,  
“Analysis of power saving and its impact on web traffic in cellular networks with continuous connectivity”, *PMC, vol. 8, no. 5, pp. 646-661, 2012.*
- [2] **N. Choungmo Fofack**, P. Nain, G. Neglia and D. Towsley,  
“Analysis of TTL-based cache networks”, *in Proc. ValueTools, Oct. 2012.*
- [3] **N. Choungmo Fofack** and Sara Alouf,  
“Modeling modern DNS caches”, *in Proc. ValueTools, Dec. 2013.*
- [R] **N. Choungmo Fofack**, P. Nain, G. Neglia and D. Towsley,  
“Performance analysis of hierarchical TTL-based cache networks”,  
*under minor revisions at Computer Networks.*
- [S] **N. Choungmo Fofack**, M. Dehghan, D. Goeckel and D. Towsley,  
“An approximate analysis of general and heterogeneous cache networks”,  
*to be submitted to Performance, Mar. 2014.*