A name resolution assisted ICN design, supported by opportunistic search, routing and caching policies *

Guilherme Domingues[1], **Edmundo de Souza e Silva**[1], **Rosa Leão**[1], **Daniel S. Menasché**[1]

[1]Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro, RJ – Brasil

Abstract. *Information centric networks (ICN), a new approach for content dissemination networks, has gained significant attention from researchers and practioners in recent years. ICN is an alternative to host-centric TCP/IP based architectures such as peer-to-peer networks (P2P) and content distribution networks (CDN). ICN performance analysis poses novel challenges. We consider a new design for an ICN relying on a third-party publishing area name resolution approach. The name resolution policy is responsible for directing requests to known content replicas inside the publishing areas, whereas the cache network offers support to opportunistic searching, routing and caching policies. That way, requests may be solved before reaching the publishing areas. In this paper, we propose a mathematical model yielding performance metrics such as mean time to find a searched content and the load experienced by the servers custodians as a function of the load issued by the users and the network topology.*

1. Introduction

Nowadays, content distribution is on focus (e.g. Youtube [Youtube 2014], Netflix [Netflix 2014], VideoAula@RNP [Land 2014]). According to [Cisco 2013], global IP traffic will reach 1,4 zettabytes per year, or 120,6 hexabytes per month by 2017, in what comprises a zettabyte era. This is manly attributed to video traffic, corresponding to approximately 80 to 90 percentage of global consumer traffic by 2017. Today, host-to-host network architectures, such as content distribution networks (CDN) and peer-to-peer networks, provide support for content dissemination.

In CDNs, *origin servers* store at least one replica of all published contents. In addition, origin servers have pointers to all content replicas stored in the network and central controllers are deployed to direct users' requests to content replicas. The most popular contents are replicated closer to the users (e.g. Akamai Networks). In P2P systems, users (also know as *peers*) act simultaneously as clients and servers (e.g. Bittorrent Networks). Despite the success of current CDNs and P2P systems, the future scalability of those architectures still poses significant challenges.

In a scenario where thousands of mobile nodes will be requesting contents, central controllers at CDNs may not scale well according to the load issued by users. In P2P networks, the throughput might not perfectly scale with respect to users' demand [de Souza e Silva et al. 2013]. A networked-centric proposal, named information centric network (ICN) [Carofiglio et al. 2013], has emerged as an alternative to deal with this scalability

*This research was supported in part by grants from CNPq and FAPERJ.

issues. In ICNs, all searched content is previously stored in the network, through subscriptions by publishers. A universal caching strategy is deployed where caching can be potentially implemented by all routers in the network.

Caching strategies in information centric networks pose new challenges [Diallo et al. 2013, Muscariello et al. 2011, Muscariello et al. 2010, Zhang et al. 2013]. In information centric networks, caching and replacement policies take place in every cache equipped device, at line-speed. The solutions deployed for content dissemination networks nowadays rely somehow on application layer solutions on top of TCP/IP networks, designed for host-to-host communications, imposing thus functional limitations for line-speed caching. Information centric network is designed for direct host to network communications encompassing routers fledged with line-speed caching capabilities.

In this paper, we consider a novel ICN architecture framework. Thus, we do not rely on add-ons onto previous ICN architecture proposals. The routers are disposed along hierarchical tiers domains, with a random lookup search strategy exploring the vicinity of caches within domains, so as to opportunistically try to reach contents being requested by users. Publishing areas provide at least one replica of each published contents at the top of the hierarchy, comprising a third party name resolution approach.

In our previous work [Domingues et al. 2013], we focused our analysis in one domain and considered a fixed cost for the publishing area. In this paper, our main contributions are: (a) the development of an analytical model which yields closed form expressions for performance metrics across the entire network, (b) the study of the tradeoffs between the delay spent on searching for content along the tiers and the total load reaching the publishing areas and (c) an evaluation of how the content replacement policy proposed in this paper (known as *reinforced counters*) reduces the delay to find popular contents, considering a given search cost at the publishing areas. The analytical model is inspired by reliability theory concepts [de Souza e Silva and Muntz 1992]. The model yields metrics such as the mean time to find a content and the total load not filtered across the network, reaching the publishing areas.

As stated in [Chai et al. 2013, Fayazbakhsh et al. 2013], ubiquitous caching across an entire network can be excessive. For this reason, our architecture does not account for a network wherein all the routers will cache contents. In fact, we consider a network assisted topology to publishing areas offering name resolution policies. Those policies are supplied by third parties at the publishing areas. We focus our attention on the infrastructure from users up to the publishing areas. The name resolution policies might be offered, for instance, by a CDN already deployed over a TCP/IP network.

The remainder of this paper is organized as follows. Section 2 presents related work and a background on Information Centric Networks. Section 3 introduces the proposed ICN system design, Section 4 contains the analytical model and in Section 5 we present numerical results obtained with the proposed model. Section 6 concludes the paper.

2. Related Work

A survey comprising many architectures being considered for ICN can be found in [Xylomenos et al. 2014]. The ICN proposed architectures can be classified into structured topology architectures, such as Dona [Koponen et al. 2007], Psirp [Fotiou et al. 2012] and NetInf [Dannewitz et al. 2013] and unstructured topology architectures, such as [Rosensweig et al. 2010]. Dona consists of an hierarchy of domains, wherein a resolution handler (RH) knows the replica storage placement of all the content published in descendent domains. RHs placed in the highest domain are aware of all the content published in the entire network. Psirp and Netinf are proposals that deploy Distributed Hash Tables (DHT). Many unsolved security vulnerabilities are able to disrupt the pre-defined operation of DHTs nodes [Urdaneta et al. 2011]. Those proposals adopts a service for resolving names to addresses. Dona uses resolution handlers to offer this service. Psirp uses rendez-vous nodes to match request and information about published contents. The service for resolving names in NetInf is called NRS (name resolution system).

NDN [Jacobson et al. 2009] propose a non structured geometry topology for routing requests. Routing tables, supporting name aggregation, route requests, towards custodian servers, leaving a trail called *bread crumbs*. Content is delivered from the custodians to the users according to the reverse paths set by the breadcrumbs. The routes can also forward content to caches across the network. Avoiding an explosion of the size of the routing tables becomes an important challenge. To enhance the discovery of cached contents, Rosensweig et al. [Rosensweig et al. 2010] allow bread crumbs not to be consumed on the fly, when content flows towards users. This preserves for some time the breadcrumb trails for previously downloaded contents. Other proposals such as [Fricker et al. 2012], [Chiocchetti et al. 2013], also based on routing content tables, try to enhance the performance of the network in what concerns the mean time to hit a searched content and the total load arriving at the custodians. Nevertheless, the updating process of content routing tables becomes also another big challenge for architectures based on content routing table designs.

In this article, we avoid the drawbacks mentioned above. To this aim, we propose a novel architecture with cache-routers disposed across hierarchical domains. Random walks are issued to explore domains' vicinity, so as to allow opportunistic encounters with the searched content. That way, we avoid the drawbacks of structured geometries, as well as the problem of fulfilling routing tables. Our design comprises a network assisted topology to publishing areas relying on a service for resolving names, offered by third parties. We do not consider ubiquitous caching across the entire network.

3. System Design

Our system comprises a logical hierarchy consisting of tiers. Each tier is divided into domains: a set of interconnected routers able to store content replicas. Each router belongs to only one domain. Figure 1 displays routers forwarding requests towards a single publishing area (green arrows). Copies of popular contents may be cached into routers within the domains. The search strategy within a domain allows *opportunistic* encounters at random (purple arrows), between users' requests flowing to the publishing area and replicas stored within the domains. This prevents certain requests to reach the publishing area. Content follows the reverse path (blue arrows) left by a request as it traverses the network. This trail left by a request is erased as the content is delivered to the users.

In this work, in contrast to our previous work [Domingues et al. 2013], we consider that all routers within a domain are virtually connected through an IP network. Each router can choose any other router in the same domain as the next hop during the random walk. In our previous work, the flow issued by the users was modeled according to a fluid model. Also, we have focused on the analysis of a single domain. We considered a fixed delay cost up to the publishing area and extra domains a request should cross, in case a searched content has not been found within the domain. In the present work, we provide a mathematical model for the entire network, comprising all domains a request should cross to hit a searched content.

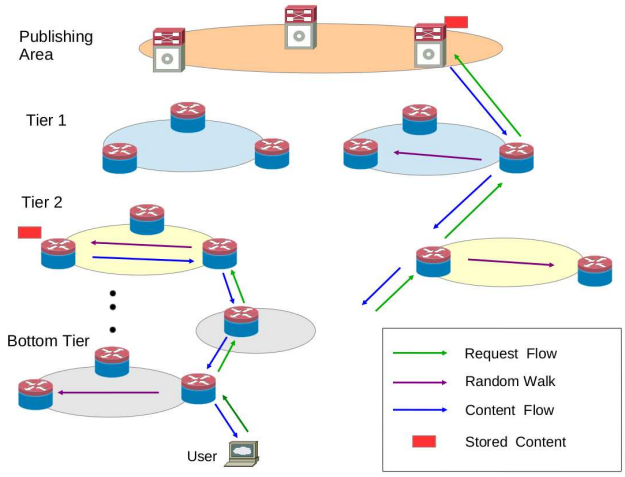


Figure 1. ICN proposed architecture

Each published content in the network is identified by a unique hash key inf. Each router in the network contains a set of counters, named reinforced counters, with two thresholds: an upper and a lower one. All counters are initialized to their lower thresholds. Whenever a request for the content inf reaches a router, either (i) an already pre-allocated counter for inf is incremented in this router by one or (ii) a free counter is allocated for inf in this router and incremented by one. The reinforced counters are decreased solely based on time. Whenever the counter for inf reaches the upper threshold in a router, the respective content for inf becomes cached while crossing this router towards the users. In contrast, whenever the counter for inf reaches the lower threshold in a router, if the content for inf is cached at this router, this content gets immediately removed and the correspondent counter is deallocated from inf. As the requests are forwarded upwards across the hierarchy, the set of *b*ackward pointers is maintained. Those pointers are named *bread crumbs*. When content is located in the network, it is delivered back to the users following the reverse path of the bread crumbs. As the content follows the path of *bread crumbs*, the trail is erased. Random walks within domains do not generate bread crumbs. This means that random walks do not impose extra hops/side effects when content is delivered to users.

4. Model

We propose an analytical model that takes into account the impact of the delay introduced when content is searched via random walk, the influence of the number of tiers and the reinforced counters. We seek to obtain the following performance goals: (i) to reduce the

variable	description
T	maximum time of a walker within each domain
$L(T)$	lifetime of random walk in a tier domain
$R(T)$	probability of not hitting content
$H(T)$	hitting time for a request reach a searched content
N	number of caches in a domain
Φ	number of domains a request should cross up to the publishing area
$T_\Phi(\lambda)$	mean time to find a content at the publishing area
λ	mean load entering the publishing area

Table 1. Notation.

total load reaching the areas where at least one copy of all published content is stored and (ii) to enhance the mean user experience time delay to find popular content. More than one replica of the searched content can be present in the domain. Within each domain, a walker explores the domain jumping from cache to cache. We consider all caches are logically fully connected. In what conforms implementation, an IP network can support the jumps of a walker between two caches not physically connected. The maximum time a walker spends in any domain is a fixed parameter T and the number of caches in a domain is a fixed parameter N.

Request for contents arrive at each domain according to a Poisson process. The probability that a request does not find the searched content in a given domain is independent of that probability in previous searched domains. We assume the time required for the processing and transmission of requests at each step of a random walk is an exponentially distributed random variable with rate γ. We model the dynamics of the reinforced counter as a birth-death process at each cache. The birth rate is given by the arrival rate Λ of requests to the cache and the death rate by an exponential random variable with rate μ, a parameter that can be set by the system tailored for best performance through proper allocation of contents along the network tier domains.

Let $p_m = m/N$ be the fraction of caches containing the desired replicas, which are present in m out of N caches in a domain. Let $R(t|J = j, M = m)$ be the probability that a walker does not find a searched content, given j transitions occurred by time t, with m desired replicas present at the domain. For the topology considered, we approximate as p_m the probability each walker succeeds on finding the desired content after each transition. Therefore, the probability that the walker does not find the desired content after j transitions is

$$R(t|J = j, M = m) \quad = \quad (1 - p_m)^j \tag{1}$$

Let γ be the rate at which a walker jumps across caches in a domain. We assume that the time between jumps (i.e. time a walker spends in a cache plus transmission time between caches) is characterized by an exponential random variable with rate γ. Then, unconditioning (1) we obtain:

$$R(t|M = m) \quad = \quad e^{-\gamma p_m t} \tag{2}$$

The derivation of (2) is found in Appendix B.

Given the reinforced counter is modeled as a birth-death process, the probability $\pi_{up}(k)$, at steady state, that the counter is greater than an upper threshold k equals to the probability that there exists, at steady state, at least k users in a $M/M/1$ queue. Let Λ be the birth rate and μ be the death rate of the counter. Let $\rho = \frac{\Lambda}{\mu}$. Then, $\pi_{up}(k)$ is given by

$$\pi_{up}(k) = \sum_{i=k}^{\infty} \rho^i \left(1 - \rho\right) \tag{3}$$

Unconditioning (2), and noting that the probability that each cache holds the content is given by (3), we obtain

$$R(t) \quad = \quad \left[\left(e^{-\gamma t/N}\right) \pi_{up}(k) + (1 - \pi_{up}(k))\right]^N \tag{4}$$

The derivation of (4) is found in Appendix C.

According to (4), $R(t) \to (1 - \pi_{up}(k))^N$ as $t \to \infty$. As the random walk time increases, the probability that the walker does not find the searched content approaches the probability that all N caches within the domain do not hold the searched content.

Let T be the maximum time a walker spends in each domain. Then, the expected lifetime of a walker within a domain is

$$E[L(T)] = \int_0^T R(t)dt \tag{5}$$

The derivation of (5) is found in [de Souza e Silva and Gail 2000].

Let Φ be the number of domains a request should cross to reach the publishing area. Let $R_i(T)$ be the probability that a search fails at domain i and $E[L_i(T)]$ the lifetime of a walker within domain i. The probability $\Psi_v(T)$ that a request has not yet hit a content after leaving domain $v = 1, 2, \ldots, \Phi$ is:

$$\Psi_v(T) = \prod_{i=1}^{v} R_i(T) \tag{6}$$

The publishing area load λ is given by $\Psi_\Phi(T)$ times the load coming from the users, $i.e.$, the publishing area load is the fraction of requests that reaches the publishing area. Let $E[H(T)]$ be the expected hitting time of a request across the entire network, accounting for the time required for the publishing area to serve the request if needed. Setting $\Psi_0 = 1$, $E[H(T)]$ is given by:

$$E[H(T)] = \left(\sum_{i=1}^{\Phi} E[L_i(T)]\Psi_{i-1}(T)\right) + T_\Phi(\lambda)\Psi_\Phi(T), \tag{7}$$

where $T_\Phi(\lambda)$ is the mean cost (measured in time units) to retrieve a content at the publishing area as a function of the mean load λ entering the publishing area.

5. Numerical Results

In this section we show the effects of load aggregation along a network architecture with multiple logical domains, comparing the multiple domain architecture with a single domain architecture. Dividing the network into multiple domains yields better performance than having a single domain. We consider scalability costs at the publishing area. Our reference topologies are illustrated in Figure 2(a) and Figure 2(b).

In the first topology, we have a single tier whereas in the second topology, three tiers. In both topologies, the same amount of caches are deployed in what concerns the number of caches connected to the users. Nonetheless, in the multi-tiered topology, there are additionally, 3000 caches placed at middle domain and 1000 caches at a single domain close to the publishing area. The multi tiered topology allows load aggregation across the tiers. The same incoming load from users is used in both topologies. The incoming load from users is modeled as a Poisson process. For the sake of analysis, it is considered a scenario with an homogeneous distributed load from users entering the tier connected to the users.

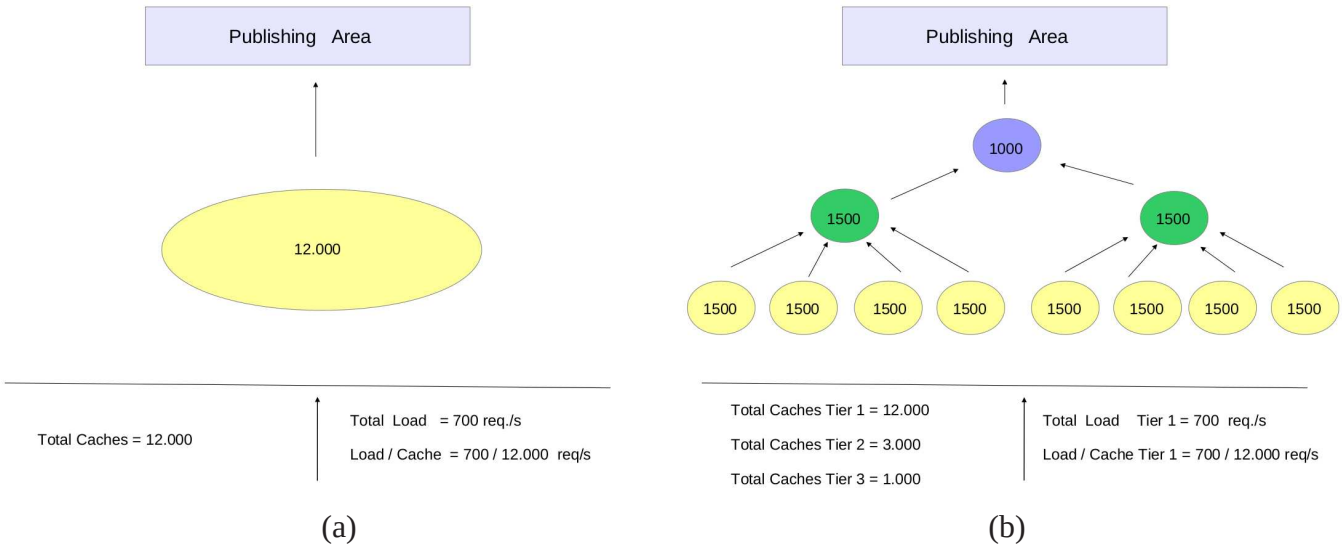


Figure 2. (a) 1-Tier Architecture (b) 3-Tiers Architecture

The mean cost time W to access the publishing area servers will be a function of the amount of traffic λ hitting the publishing area, as scalability costs are taken into account at this area. To illustrate the impact of the load of users, the publishing area search cost is modeled as an M/M/1 queue, with service capacity μ. The mean waiting time experienced by users equals to $W = 1/(\mu - \lambda)$. Figure 3 illustrates how W varies as a function of λ. It is possible to observe there is a critical value λ_* after which W starts growing exponentially.

Random walks spends a maximum time T at each domain. The probability $R(T)$ a request reaches the publish area is illustrated in Figure 4(a), as a function of T. The blue line is obtained for the architecture with a single tier. The reinforced counters parameters selected (upper threshold and decreasing rate), for the given input load from the users in the figure, enforces the probability a cache stores a content becoming negligible in the bottom tier. We can observe, in the single tier topology, that all the input flow coming

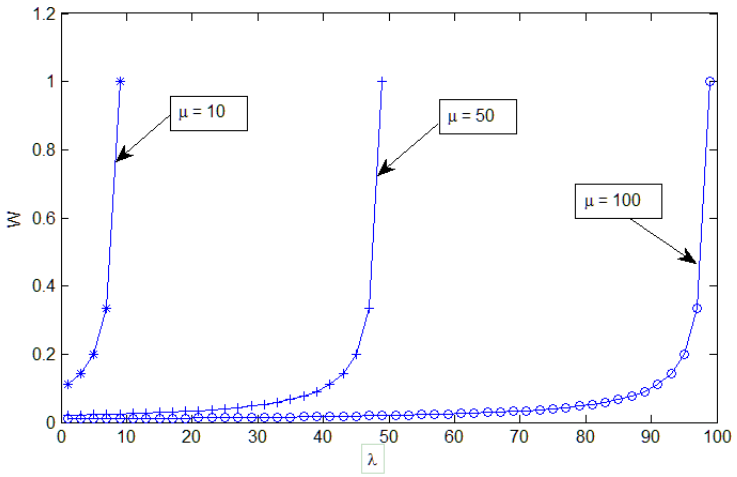


Figure 3. Average server processing time as a function of λ

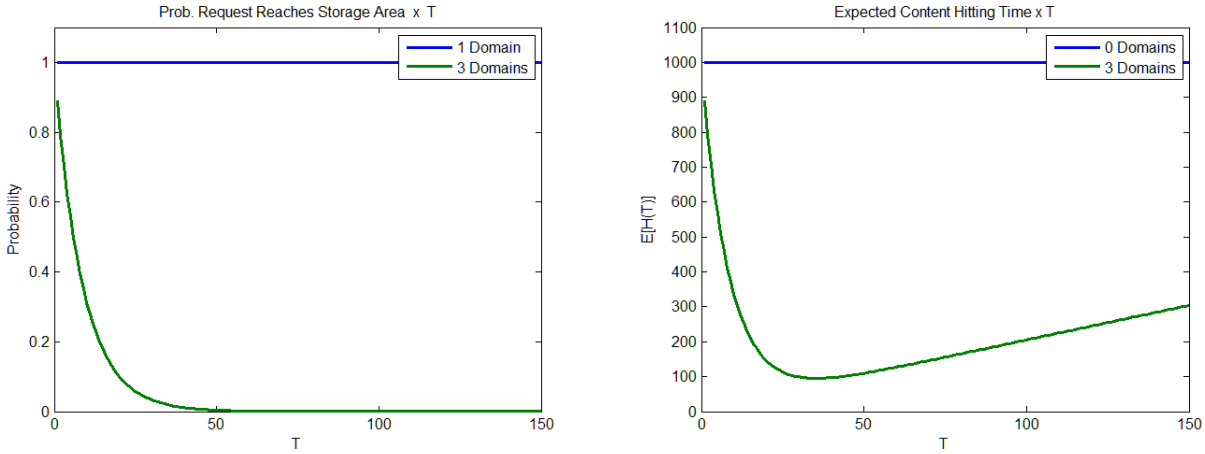
from users is redirected to the publishing area. In this scenario, no filtering effects on requests is obtained. The green line, in contrast, depicts load aggregation effects due to placing additional tiers in the multi tiered topology. The reinforced counters parameters are the same across all tiers in this topology. Also, the input load coming from users, in both topologies, are the same. Due to load aggregation effects in the upper tiers, as T increases, the probability a request does not reach the publishing area decreases almost to zero in the multi tiered topology.

The mean expected hitting time across the network is a function of the probability a request is filtered by the cache network. Henceforth, the content mean searching delay in the network benefits from load aggregation effects across the multiple tiers, when there is a non negligible searching cost within the publishing areas, as depicted in Figure 4(b). As T increases, the mean time to hit the desired content first decreases and then increases. We can see there is an upper bound for T up to which the random walk implies the mean time to hit the searched content will decrease. At the servers in the publishing area, the mean time to retrieve a content is considered to be an exponential increasing function of the load. The flat blue line in Figure 4(b) is the cost for retrieving information in the publishing area if no caches were present. In this case, the load from users would be sent directly to the publishing area. For the given input load, users would incur a mean delay of 1000 seconds at the publishing area if no caches were deployed in the network.

6. Conclusion

Information centric networks are on focus today due to network architecture changes foreseen according to future users' demands challenges. In this paper, we presented a mathematical model that allows the performance evaluation of a new proposed architecture, relying on opportunistic encounters between demanded requests and supply of contents previously published and stored.

Our model computes metrics such as mean time to find a content and probability a request reaches the publishing areas. We have focused on the performance tradeoffs of the architecture. A delay is imposed by our searching strategy, in favor of maximizing user



(a) (b)

Figure 4. (a) Prob. Request reaches Publishing Area, and (b) Mean Expected Time a request hits the searched content along the entire network

experience, under a non-cooperative cache network, with the absence of global knowledge entities, targeting reducing operational complexity costs for the architecture.

This work also opens additional avenues for future research. One such problem consists of determining the impact of parallel random walks across tiers at the same level of the hierarchy as well as across different levels. Also, the model opens the possibility for storage capacity planning whenever the amount of cache storage space turns not to be unlimited in the caches.

References

Carofiglio, G., Morabito, G., Muscariello, L., Solis, I., and Varello, M. (2013). From content delivery today to information centric networking. *Computer Networks*, 57:3116–3127.

Chai, W. K., He, D., Psaras, I., and Pavlou, G. (2013). Cache ”less for more” in information-centric networks. *Computer Communications*, 36:758–770.

Chiocchetti, R., Perino, D., Carofiglio, G., Rossi, D., and Rossini, G. (2013). Inform: a dynamic interest forwarding mechanism for information centric networking. *Proc. of 3rd ACM SIGCOMM Workshop on ICN*.

Cisco (2013). Cisco visual networking index, forecast and methodology, 2012 - 2017. *White Paper*.

Dannewitz, C., Kutscher, D., Ohlman, B., Farrell, S., Ahlgren, B., and Kar, H. (2013). Network of information (netinf) – an information-centric networking architecture. *Elsevier Computer Communications*, 36:721–735.

de Souza e Silva, E. and Gail, H. R. (2000). Transient Solutions for Markov Chains. In Grassmann, W., editor, *Computational Probability*, pages 44–79. Kluwer.

de Souza e Silva, E., Leão, R. M. M., Menasché, D. S., and de A. Rocha, A. A. (2013). On the interplay between content popularity and performance in p2p systems. *QEST*, pages 27–30.

de Souza e Silva, E. and Muntz, R. R. (1992). *Métodos Computacionais de Solução de Cadeias de Markov: Aplicações a Sistemas de Computação e Comunicação*. SBC - Escola de Computação.

Diallo, M., Sourlas, V., Flegkas, P., Fdida, S., and L.Tassiulas (2013). A content-based publish/subscribe framework for large-scale content delivery. *Computer Networks*, 57:924–943.

Domingues, G. M. B., de Souza e Silva, E. A., Leão, R. M. M., and Mensché, D. S. (2013). Enabling information centric networks through opportunistic search, routing and caching. In *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.

Fayazbakhsh, S. K., Lin, Y., Tootoonchian, A., Godshi, A., Koponen, T., Maggs, B. M., Ng, K., Sekar, V., and Schenker, S. (2013). Less pain, most of the gain: Incrementally deployable icn. *ACM SIGCOMM*, pages 12–16.

Fotiou, N., Nikander, P., Trossen, D., and Polyzos, G. C. (2012). Developing information networking further: From psirp to pursuit. In *Broadband Communications, Networks, and Systems, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, volume 66, pages 1–13.

Fricker, C., Robert, P., Roberts, J., and Sbihi, N. (2012). Impact of traffic mix on caching performance in a content-centric network. *Proc. of IEEE NOMEN 2012*.

Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. (2009). Networking named content. In *Proc. of CoNEXT*, pages 1–12.

Koponen, T., Chawla, M., Chun, B. G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I. (2007). A data-oriented (and beyond) network architecture. In *Proc. of SIGCOMM*, pages 181–192.

Land (2014). Land. `http://www.land.ufrj.br/edad`.

Muscariello, L., G.Carofiglio, and M.Gallo (2010). Greening the internet with content-centric networking. *eEnergy*, 179.

Muscariello, L., G.Carofiglio, and M.Gallo (2011). Bandwidth and storage sharing performance in information centric networks. *ACM SIGCOMM ICN Workshop*, page 2631.

Netflix (2014). Netflix. `http://www.netflix.com`.

Rosensweig, E., Kurose, J., and Towsley, D. (2010). Approximate models for general cache networks. In *Proc. of INFOCOM*, pages 1–9.

Urdaneta, G., Pierre, G., and Steen, M. V. (2011). A survey of DHT security techniques. *ACM Comp. Surveys*, 43(2).

Xylomenos, G., Ververidis, C. N., Siris, V. A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K. V., and Polyzos, G. C. (2014). A survey of information-centric networking research. *Communications Surveys and Tutorials*.

Youtube (2014). Youtube. `http://www.youtube.com`.

Zhang, G., Li, Y., and Lin, T. (2013). Caching in information centric networking: A survey. *Computer Networks*, 57:3128–3141.

Appendix A: Terminology

In this appendix we summarize the terminology adopted throughout the paper.

- *cache-router:* a router augmented with a cache.
- *domain:* logical set of cache-routers.
- *tier:* a set of domains. Each domain is contained in a tier. Tier N is the bottom tier, and receives user requests. The top tier is connected to the servers.
- *server:* element that store fixed copies of the content (also known as custodian in the literature of cache networks).
- *message:* a request or a search message.
- *request:* a message that goes from users to tier N, or inter-tiers, or from the top-tier to the servers.
- *search message:* a random walk message that goes intra-tier.
- *bread crumb:* a piece of information about a content consisting of a backward pointer and a hash key of the content that it refers to.
- *reinforced counter:* a counter that determines whether content should be stored at a given cache-router.

Appendix B: Derivation of (2)

In this appendix, we show how to obtain $R(t|M = m)$ from (1). As stated in Section 4, $R(t|J = j, M = m)$ equals to the probability that a walker does not find a searched content, given j transitions occurred by time t, with m desired replicas present at the domain. We consider a fast walker such that m does not change during the entire walk. Let γ be the rate at which a walker jumps across caches in a domain. Unconditioning $R(t|J = j, M = m)$, given that the distribution of J follows a Poisson distribution with parameter γ, we obtain:

$$
\begin{aligned}
R(t|M = m) &= \sum_{j=0}^{\infty} \frac{(\gamma t)^j}{j!}(1 - p_m)^n e^{-\gamma t} && (8)\\
&= \frac{1}{e^{\gamma t p_m}} \sum_{n=0}^{\infty} \frac{(\gamma t(1 - p_m))^j}{j!} e^{-\gamma t} e^{\gamma t p_m} && (9)\\
&= \frac{1}{e^{\gamma t p_m}} \sum_{n=0}^{\infty} \frac{(\gamma t(1 - p_m))^j}{j!} e^{-\gamma t(1-p_m)} && (10)\\
&= e^{-\gamma p_m t} && (11)
\end{aligned}
$$

Appendix C: Derivation of (4)

In this appendix, we show how to obtain $R(t)$ from (2). As stated in Section 4, the reinforced counter is modeled as a birth-death process, and $\pi_{up}(k)$, at steady state, equals to the probability the counter is greater than an upper threshold k. Unconditioning $R(t|M = m)$, given that the probability distribution of M follows a Binomial distribution with parameter $\pi_{up}(k)$, we obtain:

$$
\begin{aligned}
R(t) &= \sum_{m=0}^{N} R(t|M=m)P(M=m) && (12)\\
&= \sum_{m=0}^{N} R(t|M=m)\binom{N}{m}(\pi_{up}(k))^m(1-\pi_{up}(k))^{N-m} && (13)\\
&= \sum_{m=0}^{N} e^{-\gamma p_m t}\binom{N}{m}(\pi_{up}(k))^m(1-\pi_{up}(k))^{N-m} && (14)\\
&= \sum_{m=0}^{N} \binom{N}{m}\left(e^{-\gamma m t/N}\right)(\pi_{up}(k))^m(1-\pi_{up}(k))^{N-m} && (15)\\
&= \sum_{m=0}^{N} \binom{N}{m}\left[\left(e^{-\gamma t/N}\right)\pi_{up}(k)\right]^m\left[(1-\pi_{up}(k))^{N-m}\right] && (16)\\
&= \left[\left(e^{-\gamma t/N}\right)\pi_{up}(k)+(1-\pi_{up}(k))\right]^N && (17)
\end{aligned}
$$