On the Throughput Capacity of Information-Centric Networks

Bita Azimdoost[†], Cedric Westphal[‡*], and Hamid R. Sadjadpour[†]
[†]Department of Electrical Engineering and [‡]Computer Engineering
University of California Santa Cruz, Santa Cruz, CA 95064, USA
{bazimdoost,cedric,hamid}@soe.ucsc.edu
[*] Huawei Innovation Center, Santa Clara, CA 95050, USA
cedric.westphal@huawei.com

***Abstract*—Wireless information-centric networks consider storage one of the network primitives, and propose to cache data within the network in order to improve latency to access content and reduce bandwidth consumption. We study the throughput capacity of an information-centric network when the data cached in each node has a limited lifetime. The results show that with some fixed request and cache expiration rates, the network can have the maximum throughput order of $1/\sqrt{n}$ and $1/\log n$ in cases of grid and random networks, respectively. Comparing these values with the corresponding throughput with no cache capability ($1/n$ and $1/\sqrt{n \log n}$ respectively), we can actually quantify the asymptotic advantage of caching. Moreover, since the request rates will decrease as a result of increasing download delays, increasing the content lifetimes according to the network growth may result in higher throughput capacities.**

I. Introduction

In today's networking situations, users are mostly interested in accessing content regardless of which host is providing this content. They are looking for a fast and secure access to data in a whole range of situations: wired or wireless; heterogeneous technologies; in a fixed location or when moving. The dynamic characteristics of the network users makes the host-centric networking paradigm inefficient. Information-centric networking (ICN) is a new networking architecture where content is accessed based upon its name, and independently of the location of the hosts [1]–[4]. In most ICN architectures, data is allowed to be stored in the nodes and routers within the network in addition to the content publisher's servers. This reduces the burden on the servers and on the network operator, and shortens the access time to the desired content.

Combining content routing with in-network-storage for the information is intuitively attractive, but there has been few works considering the impact of such architecture on the capacity of the network in a formal or analytical manner. In this work we study a wireless information-centric network where nodes can both route and cache content. We also assume that a node will keep a copy of the content only for a finite period of time, that is until it runs out of memory space in its cache and has to rotate content, or until it ceases to serve a specific content.

The nodes issue some queries for content that is not locally available. We suppose that there exists a server which permanently keeps all the contents. This means that the content is always provided at least by its publisher, in addition to the potential copies distributed throughout the network. Therefore, at least one replica of each content always exists in the network and if a node requests a piece of information, this data will be provided either by its original server or by a cache containing the desired data. When the customer receives the content, it will store the content and share it with the other nodes if needed.

The present paper thus investigates the throughput capacity in such content-centric networks and addresses the following questions:

1) Looking at the throughput capacity, can we quantify the performance improvement brought about by a content-centric network architecture over networks with no content sharing capability?
2) How does the content discovery mechanism affect the throughput capacity? More specifically, does selecting the nearest copy of the content improve the scaling of the capacity compared to selecting the nearest copy in the direction of original server?
3) How does the caching policy, and in particular, the length of time each piece of content spends in the cache's memory, affect the capacity?

We state three Theorems; Theorem 1 formulates the throughput capacity in a grid network which uses the shortest path to the server content discovery mechanism considering different content availability in different caches, and Theorem 2 will answer the first two questions studying two different network models (grid and random network) and two content discovery scenarios (shortest path to the server and shortest path to the closest copy of the content) when the information exists in all caches with the same probability. Theorem 3 derives some conditions on the respective request rate (namely, the popularity of the content) and the time spent in the cache, so that these throughputs can be supported by all the nodes and the flow in no node be a bottleneck. These theorems demonstrate that adding the content sharing capability to the nodes can significantly increase the capacity.

Bita Azimdoost was with Huawei Innovation Center, Santa Clara, CA 95050, USA, as an intern while working on this paper.

The rest of the paper is organized as follows. After a brief review of the related work in Section II, the network models and the content discovery algorithms used in the current work are introduced in Section III. Main Theorems are stated and proved in Section IV. We will discuss the results and study some simple examples in Section V. Finally the paper is concluded and some possible directions for the future work will be introduced in section VI.

II. Related Work

Information Centric Networks have recently received considerable attention. While our work presents an analytical abstraction, it is based upon the principles described in some ICN architectures, such as CCN [4], NetInf [5], PURSUIT [2], or DONA [6], where nodes can cache content, and requests for content can be routed to the nearest copy. Papers surveying the landscape of ICN [3] [7] show the dearth of theoretical results underlying these architectures.

Caching, one of the main concepts in ICN networks, has been studied in prior works [3]. Some performance metrics like miss ratio in the cache, or the average number of hops each request travels to locate the content have been studied in [8], [9], and the benefit of cooperative caching has been investigated in [10].

Optimal cache locations [11] and cache replacement techniques [12] are two other aspects most commonly investigated. And an analytical framework for investigating properties of these networks like fairness of cache usage is proposed in [13]. [14] considered information being cached for a limited amount of time at each node, as we do here, but focused on flooding mechanism to locate the content, not on the capacity of the network.

However, to the best of our knowledge, there are just a few works focusing on the achievable data rates in such networks. Calculating the asymptotic throughput capacity of wireless networks with no cache has been solved in [15] and many subsequent works [16] [17]. Some work has studied the capacity of wireless networks with caching [18] [19] [20]. There, caching is used to buffer data at a relay node which will physically move to deliver the content to its destination, whereas we follow the ICN assumption that caching is triggered by the node requesting the content.

[21] uses a network simulation model and evaluates the performance (file transfer delay) in a cache-and-forward system with no request for the data. [22] proposes an analytical model for single cache miss probability and stationary throughput in cascade and binary tree topologies. [23] considers a general problem of delivering content cached in a wireless network and provides some bounds on the caching capacity region from an information-theoretic point of view. Some scaling regimes for the required link capacity is computed in [24] for a static cache placement in a multihop wireless network.

III. Preliminaries

A. Network Model

Two network models are studied in this work.

1) Grid Network: Assume that the network consists of n nodes $V = \{v_1, v_2, ..., v_n\}$ each with a local cache of size L located on a grid (Figure 1). The distance between two adjacent nodes equals to the transmission range of each node, so the packets sent from a node are only received by four adjacent nodes. There are m different contents, $F = \{f_1, f_2, ..., f_m\}$ with sizes $B_i,\ i = 1, ..., m$, for which each node v_j may issue a query. Based on the content discovery algorithms which will be explained later in this section, the query will be transmitted in the network to discover a node containing the desired content locally. v_j then downloads b bits of data with rate γ in a hop-by-hop manner through the path P_{xj} from either a node ($v_i, x = i$) containing it locally ($f \in v_i$) or the server ($x = s$). When the download is completed, all the nodes on the download path or just the end user store the data in their local cache and share it with other nodes. P_{js} denotes the nodes on the path from v_j to server. Without loss of generality, we assume that the server is attached to the node located at the middle of the network, changing the location of the server does not affect the scaling laws. Using the protocol model and according to [25] the transport capacity in such network is upper bounded by $\Theta(W\sqrt{n})$. This is the model studied in 1 and the first two scenarios of Theorem 2.

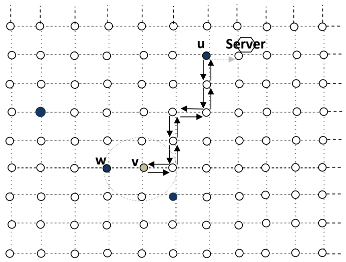

Fig. 1. The transmission range of node v contains four surrounding nodes. The black vertices contain the content in their local caches. The arrow lines demonstrate a possible discovery and receive path in scenario i, where node v downloads the required information from u. In scenario ii, v will download the data from w instead.

2) Random Network: The last network studied in Theorem 2 is a more general network model where the nodes are randomly distributed over a unit square area according to a uniform distribution. We use the same model used in [25] (section 5) and divide the network area into square cells each with side-length proportional to the transmission range $r(n)$, which is selected to be at least in the order of $\sqrt{\frac{\log n}{n}}$ to guarantee the connectivity of the network [26]. According to the protocol model [25], if the cells are far enough they can transmit data at the same time with no interference; we assume that there are M^2 non-interfering groups which take turn to transmit at the corresponding time-slot in a round robin fashion. The server is assumed to be located at the middle of the network. In this model the maximum number of simultaneous feasible transmissions will be in the order of $\frac{1}{r^2(n)}$ as each transmission consumes an area proportional to

$r^2(n)$.

All the other assumptions are similar to the grid network.

B. Content Discovery Algorithm

1) Path-wise Discovery: To discover the location of the desired content, the request is sent through the shortest path toward the server containing the requested content. If an intermediate node has the data in its local cache, it does not forward the request toward the server anymore and the requester will start downloading from the discovered cache. Otherwise, the request will go all the way toward the server and the content is obtained from the main source. In case of the random network when a node needs a piece of information, it will send a request to its neighbors toward the server, i.e. the nodes in the same cell and one adjacent cell in the path toward the server, if any copy of the data is found it will be downloaded. If not, just one node in the adjacent cell will forward the request to the next cell toward the server.

2) Expanding Ring Search: In this algorithm the request for the information is sent to all the nodes in the transmission range of the requester. If a node receiving the request contains the required data in its local cache, it notifies the requester and then downloading from the discovered cache is started. Otherwise, all the nodes that receive the request will broadcast the request to their own neighbors. This process continues until the content is discovered in a cache and the downloading follows after that.

IV. Theorem Statements and Proofs

Theorem 1. *Consider a grid wireless network consisting of n nodes. Each node can transmit over a common wireless channel, with W bits per second bandwidth, shared by all nodes. Assume that there is a server which contains all the information. Without loss of generality we assume that this server is located in the middle of the network. Each node contains some information in its local cache. Assume that the probability of the information being in all the caches with the same distance (j hops) from the server is the same ($\rho_j(n)$). The maximum achievable throughput capacity order[1] (γ_{max}) in such network when the nodes use the nearest copy of the required content on the shortest path toward the server is given by*

$$\Theta(\frac{W\sqrt{n}}{\sum_{i=1}^{\sqrt{n}} i \sum_{j=0}^{i}(i-j)\rho_j(n)\prod_{k=j+1}^{i}(1-\rho_k(n))}),$$

where $\rho_0(n) = 1$, which means that the server always contains the information.

Proof: A request initiated by a user v_i in i-hop distance from the server (located in level $i = 1, .., \sqrt{n}$) is served by cache u_j located in level j, $1 \leq j \leq i$ on the shortest path from v_i to the server if no caches before u_j, including v_i, on this path contains the required information, and u_j contains it. This request is served by the server if no copy of it is available on the path. Assuming that the availability of the information in each cache is independent of the contents in the other caches, this probability denoted by $P_{i,j}$ is given by

$$P_{i,j} = (1-\rho_i(n))(1-\rho_{i-1}(n))...(1-\rho_{j+1}(n))\rho_j(n) \tag{1}$$

where $\rho_j(n)$ is the probability of the information being available in a cache in level j, $j = 1, .., \sqrt{n}$, and $j = 0$ shows the server and $\rho_0(n) = 1$. Thus a content requested by v_i is traveling $i - j$ hops with probability $P_{i,j}$. There are $4i$ nodes in level i so the average number of hops ($\bar{h}$) traveled by each piece of data from the serving cache (or the original server) to the requester is

$$\begin{aligned} \bar{h} &= \frac{1}{n}\sum_{i=1}^{\sqrt{n}} 4i \sum_{j=0}^{i}(i-j)P_{i,j} \\ &= \frac{1}{n}\sum_{i=1}^{\sqrt{n}} 4i \sum_{j=0}^{i}(i-j)(1-\rho_i(n))...(1-\rho_{j+1}(n))\rho_j(n) \end{aligned} \tag{2}$$

Assume that each user is receiving data with rate γ. The transport capacity in this network, which equals to $n\gamma\bar{h}$, is upper bounded by $\Theta(W\sqrt{n})$. So $\gamma_{max} = \Theta(\frac{W}{\bar{h}\sqrt{n}})$ and the Theorem is proved. ■

Theorem 2. *Consider a wireless network consisting of n nodes, with each node containing the information in its local cache with common probability $\rho(n)$. Each node can transmit over a common wireless channel, with W bits per second bandwidth, shared by all nodes.*

- *Scenario i- If the nodes are located on a grid and search for the contents just on the shortest path toward the server, the maximum achievable throughput capacity order γ_{max} is*

$$\begin{cases} \Theta(\frac{W\rho(n)}{\sqrt{n}(1-\rho(n))}) & ,if\ \rho(n) = \Omega(n^{-1/2}) \\ \Theta(\frac{W}{n(1-\rho(n))}) & ,if\ \rho(n) = O(n^{-1/2}) \end{cases}$$

- *Scenario ii- If the nodes are located on a grid and use ring expansion as their content search algorithm, the maximum achievable throughput γ_{max} is*

$$\begin{cases} \Theta(\frac{W\rho(n)^{0.4646}}{\sqrt{n}(1-\rho(n))}) & ,if\ \rho(n) = \Omega(n^{-1/2}) \\ \Theta(\frac{W}{n(1-\rho(n))}) & ,if\ \rho(n) = O(n^{-1/2}) \end{cases}$$

- *Scenario iii- If the nodes are randomly distributed over a unit square area and use path-wise content discovery algorithm, the maximum achievable capacity γ_{max} is*

$$\begin{cases} \Theta(\frac{W}{\log n(1-\rho(n))}) & ,if\ \rho(n) = \Omega(\frac{1}{\log n}) \\ \Theta(\frac{W}{\sqrt{n \log n}(1-\rho(n))}) & ,if\ \rho(n) = O(\frac{1}{\log n}) \end{cases}$$

Here we prove Theorem 2 by utilizing some lemmas.

[1] $f(n) = O(g(n))$ if $sup_n(f(n)/g(n)) < \infty$. $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$. $f(n) = \Theta(g(n))$ or $f(n) \equiv g(n)$ if both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$. $f(n) = o(g(n))$ if $f(n)/g(n) \to 0$. $f(n) = \omega(g(n))$ if $g(n)/f(n) \to 0$.

Lemma 1. Consider the wireless networks described in Theorem 2. For sufficiently large networks and when $\rho(n)$ is large enough ($\rho(n) = \Omega(n^{-1/2})$ for case i, ii and $\rho(n) = \Omega(\log^{-1} n)$ for case iii), the average number of hops between the customer and the nearest cached content location is

$$\bar{h} \quad = \quad \begin{cases} \Theta(\frac{1}{\rho(n)}) & (i) \\ \Theta(\frac{1}{\rho(n)^{0.4646}}) & (ii) \\ \Theta(1) & (iii) \end{cases} \tag{3}$$

Proof: Scenario i- This case can be considered as a special case of the network studied in theorem 1, where $\rho_i(n)$ is the same for all i. Thus we can drop the index i and let $\rho(n)$ denote the common value of this probability. Using equation 2 we will have

$$\begin{aligned} & \bar{h} \\ = \; & \frac{4}{n}\sum_{i=1}^{\sqrt{n}} i\{i(1-\rho(n))^i + \sum_{j=1}^{i}(i-j)(1-\rho(n))^{i-j}\rho(n)\} \\ = \; & \frac{4}{n}(\sum_{i=1}^{\sqrt{n}} i^2(1-\rho(n))^i + \sum_{i=1}^{\sqrt{n}} i \sum_{k=0}^{i-1} k(1-\rho(n))^k\rho(n)) \\ = \; & \frac{4(1-\rho(n))}{n\rho(n)} \sum_{i=1}^{\sqrt{n}}(i - i(1-\rho(n))^i) \\ = \; & \frac{4(1-\rho(n))}{n\rho(n)}(\frac{\sqrt{n}(\sqrt{n}+1)}{2} - \frac{1-\rho(n)}{\rho^2(n)} \\ - \; & \frac{\sqrt{n}(1-\rho(n))^{\sqrt{n}+2} - (\sqrt{n}+1)(1-\rho(n))^{\sqrt{n}+1}}{\rho^2(n)}) \end{aligned}$$

Since for every N and x the following is true

$$\lim_{N\to\infty}(1-x)^N = \begin{cases} 1 & x = o(\frac{1}{N}) \\ e^{-xN} & x = \Theta(\frac{1}{N}) \\ 0 & x = \omega(\frac{1}{N}) \end{cases}$$

we can write $(1-\rho(n))^{\sqrt{n}} \to 0$ if $\rho(n) = \Omega(n^{-1/2})$, which results in $\bar{h} = \Theta(\frac{1}{\rho(n)})$, and $(1-\rho(n))^{\sqrt{n}} \to 1$ if $\rho(n) = O(n^{-1/2})$, which results in $\bar{h} = \Omega(\sqrt{n})$, and since is upper limited by $\Theta(\sqrt{n})$ it is equal to that value.

Scenario ii - The probability that the discovered cache is located at a distance of one hop from the requester is the probability that one of the nodes on the ring at one hop distance contains the data (it consists of 4 nodes), which equals to $1-(1-\rho(n))^4$, and the probability that the data needs to travel through h hops from the discovered cache to where it is required is $(1-(1-\rho(n))^{4h})\prod_{k=1}^{h-1}(1-\rho(n))^{4k}$ as there are $4h$ nodes at distance of h hops. Therefore,

$$\bar{h} \quad = \quad \sum_{h=1}^{\sqrt{n}} h(1-(1-\rho(n))^{4h})\prod_{k=1}^{h-1}(1-\rho(n))^{4k}$$

Numerical analysis show that the above equation is $\Theta(1/\rho^{\delta})$ where $0.4 < \delta < 0.5$ if $\rho = \Omega(n^{-1/2})$. In the rest of paper we use $\delta = 0.4646$ which is obtained by curve fitting. For smaller $\rho(n)$'s, $\bar{h}$ will increase to $\Theta(\sqrt{n})$.

Scenario iii - The discovered cache is one hop away from the requester if there is a replica of the data in a cache at the same cell or at the adjacent cell toward the server. Since there are $\log n$ nodes in each cell, the probability of the discovered cache being at one hop distance is $1-(1-\rho(n))^{2\log n}$, and the probability of the discovered cache being at distance of h hops away from the requester is $(1-\rho(n))^{h\log n}(1-(1-\rho(n))^{\log n})$. The maximum number of hops that may be traveled this way is $\frac{1}{r(n)}$. Thus

$$\begin{aligned} \bar{h} &= 1-(1-\rho(n))^{2\log n} \\ &+ \textstyle\sum_{h=2}^{\frac{1}{r(n)}} h(1-\rho(n))^{h\log n}(1-(1-\rho(n))^{\log n}) \\ &\overset{Large\ n}{\cong} \Theta(1) \end{aligned} \tag{4}$$

where the last equality is correct when $\rho(n) = \Omega(\log^{-1} n)$. ■

In scenario iii the average number of hops between the nearest content location and the customer is just $\Theta(1)$ hop. This is the result of having $log(n)$ caches in one hop distance for every requester. Each one of these caches can be a potential source for the content. When the network grows, this number will increase and if $\rho(n)$ is large enough ($\rho(n) = \Omega(\log^{-1} n)$) the probability that at least one of these nodes contain the required data will approach 1, i.e., $\lim_{n\to\infty}(1-(1-\rho(n))^{\log n}) = 1$.

Lemma 2. The average probability that the server needs to serve a request is

$$p_s \quad = \quad \begin{cases} \Theta\left(\frac{(2-\rho(n))^2}{n\rho(n)^2}\right) & (i) \\ O\left(\frac{(2-\rho(n))^2}{n\rho(n)^2}\right) & (ii) \\ \Theta\left(\frac{(2-\rho(n))\log n}{n}\right) & (iii) \end{cases} \tag{5}$$

Proof: Scenario i- The data will need to be downloaded from the server (at average distance $\bar{h}_s$) if no copy of the data is available on the path between a requester node and the server. As the network area is assumed to be a square and the server is in the middle of it, this probability is bounded by

$$\frac{1+\sum_{k=1}^{h_{max}/2} 4k(1-\rho(n))^k}{n} \le p_s \le \frac{1+\sum_{k=1}^{h_{max}} 4k(1-\rho(n))^k}{n}$$

Thus for large n, $p_s = \Theta(\frac{(2-\rho(n))^2}{n\rho(n)^2})$.

Note that both $\bar{h}_s$ and h_{max}, the maximum number of hops which may be traveled between the requester and the node that possesses a valid copy of data, in this scenario are $\Theta(\sqrt{n})$.

Scenario ii- The data will need to be downloaded from the server (at average distance $\bar{h}_s$) if no copy of the content is available in the network caches. Since comparing to scenario i more nodes will be involved in the process of content discovery, it is obvious that in this case the request will be forwarded to the server with less probability. Thus $p_s = O(\frac{(2-\rho(n))^2}{n\rho(n)^2})$.

Scenario iii- The data is downloaded from the server if no node in the cells on the path toward the server cell contains a copy of the content.

$$p_s = \frac{1+5\log n(1-\rho(n))+\sum_{h=2}^{\frac{1}{r(n)}} 4h\log n(1-\rho(n))^{(h-1)\log n}}{n}$$
$$\overset{Large\ n}{\cong} \frac{(2-\rho(n))\log n}{n} \qquad (6)$$

■

It can be seen that in all cases the average number of hops between the server and the node requesting the content is a function of the total number of nodes in the network and $\rho(n)$.

Now we can prove *Theorem 2* using the above lemmas.

Proof: Assume that each content is retrieved with rate γ bits/sec. The traffic generated because of one download from a cache at average distance of $\bar{h}$ hops from the requester node is $\gamma\bar{h}$ and the traffic generated due to the downloads from the server at average distance of $\bar{h}_s$ hops from the requester is $\gamma\bar{h}_s$. The probability that the server is uploading the data is p_s and the probability that a cache node is serving the customer is $p = 1 - p_s$. The total number of requests for a content in the network at any given time is limited by the number of nodes not having the content in their own cache ($(1-\rho(n))n$). Thus the maximum total bandwidth needed to accomplish these downloads will be $(1-\rho(n))n(p\bar{h}+p_s\bar{h}_s)\gamma$, which is upper limited by $(\Theta(W\sqrt{n}))$ in scenarios i, ii, and $(\Theta(\frac{W}{r^2(n)}))$ in scenario iii.

Therefore the maximum download rate is

$$\gamma_{max} = \begin{cases} \Theta(\frac{W\sqrt{n}/n(1-\rho(n))}{(1-\frac{(2-\rho(n))^2}{n\rho(n)^2})\rho(n)^{-1}+\frac{(2-\rho(n))^2}{n\rho(n)^2}\sqrt{n}}) & (i) \\ \Theta(\frac{W\sqrt{n}/n(1-\rho(n))}{(1-\frac{(2-\rho(n))^2}{n\rho(n)^2})\rho(n)^{-0.4646}+\frac{(2-\rho(n))^2}{n\rho(n)^2}\sqrt{n}}) & (ii) \\ \Theta(\frac{W/r^2(n)n(1-\rho(n))}{(1-\frac{(2-\rho(n))\log n}{n})+\frac{(2-\rho(n))\log n}{nr(n)}}) & (iii) \end{cases} \qquad (7)$$

The results of Theorem 1 can be derived by approximating these equations for sufficiently large n. Note that if there were no cache in the system, or $\rho(n)$ is less than the stated threshold values, all the requests would be served by the server, and the maximum download rate would be $\frac{W}{\sqrt{n}\bar{h}_s} = \Theta(\frac{W}{n})$ in case i, ii and $\Theta(\frac{W}{\sqrt{n\log n}})$ in case iii. ■

In the previous Theorems the maximum throughput capacity in a cache wireless network has been calculated. Now it is important to verify if this throughput can be supported by each cell (node), i.e. the traffic carried by each cell (node) is not more than what it can support ($\Theta(1)$).

Theorem 3. *The throughput capacities of Theorem 2 are supportable if* $\rho(n) = O(\frac{n\log\log n}{\log n+n\log\log n})$ *in scenario i, ii, and* $\rho(n) = O(\frac{\log n\log\log\log n-\log\log n}{\log n\log\log\log n})$ *in scenario* iii.

Here we start with scenario iii and a complete proof. Scenario i will be then briefly studied. Similar reasoning can be used for scenario ii.

Proof: Scenario iii- The traffic load at the server is $\gamma_{max}p_sn(1-\rho(n)) = \Theta(1)$. So the flow at the server will not be a bottleneck.

The traffic load at a node as a customer will not be a bottleneck either as it does not exceed the maximum data rate which is $\gamma_{max} = \Theta(\frac{W}{\log n(1-\rho(n))}) < \Theta(1)$.

To compute the traffic load at a node which is serving a request, we need to know how many requests that node may serve at a time. A node $v_i \in V$ is the download source if it has the information ($\rho(n)$), it is in the same cell as the requester or in a cell on the path from the requester to the server and no node in the previous cells on this path contains the content ($(1-\rho(n))^{x\log n}$ where x is the number of hops between v_j and v_i), and among those nodes in the same cell which have the data v_i is selected to serve the query ($\sum_{k=1}^{\log n}\frac{1}{k}\binom{\log n-1}{k-1}\rho(n)^{k-1}(1-\rho(n))^{\log n-k}$). For not too small $\rho(n)$ and large n, we have

$$P(v_i\ is\ serving\ v_j{}'s\ request) = \begin{cases} \frac{1-\rho(n)}{\log n} & v_j,\ v_i\ \in same\ cell \\ \frac{(1-\rho(n))^{\log n}}{\log n} & h_{ji} = 1\ \&\ v_i \in P_{js} \\ \frac{(1-\rho(n))^{h+\log n}}{\log n} & (\substack{1<h_{ji}=h\leq\sqrt{\frac{n}{\log n}} \\ \&\ v_i\in P_{js}}) \\ 0, & otherwise \end{cases} \qquad (8)$$

Therefore, each node containing the content will serve only the nodes in the same cell with high probability, and the probability of being selected to serve the query initiated at the same cell is $\frac{1}{\rho(n)\log n}$. Based on the bin-balls Theorem [27], the maximum number of queries served by a node will be $\frac{\log\log n}{\log\log\log n}$. Consequently, the maximum traffic load per source is $\gamma_{max}\frac{\log\log n}{\log\log\log n} = \frac{W\log\log n}{(1-\rho(n))\log n\log\log\log n}$. Therefore, to make sure that all the cells can support the stated throughput $\rho(n)$ is not allowed to exceed $O(\frac{\log n\log\log\log n-\log\log n}{\log n\log\log\log n})$.

Finally each download of information will generate a traffic load on all the intermediate cells on the path from the source to the customer. However as stated in the proof of Theorem 2, the probability that the required content is discovered at distance of one hope is $1-(1-\rho(n))^{2\log n}$ which is almost one for large n. So we may conclude that with high probability in sufficiently large networks no cell is working as relay or the number of transmissions passing through a cell as relay is close to zero.

Scenario i- Similar to scenario iii, the maximum traffic load is the load generated in a node when serving the requests. Here there are $n(1-\rho(n))$ requests which will be served by $n\rho(n)$ other nodes, so according to the bin-balls Theorem the maximum requests for a node will be in the order of $\frac{\log n}{\log\log n}$, which generates $\frac{W\rho(n)\log n}{n(1-\rho(n))\log\log n}$traffic at the busiest node. This traffic does not exceed $\Theta(1)$ as long as $\rho(n)$ does not exceed $O(\frac{n\log\log n}{\log n+n\log\log n})$. ■

V. Discussion

We studied the impact of caching on the maximum capacity order in the grid and random networks where all the caches have the same probability of having each item at any given time. The networks where the received data is stored only at the receivers and then shared with the other nodes as long as the node keeps the content can be considered as an example of such networks. In Figure 2 (a) we assume that the request rate is roughly 7 times the drop rate, so $\rho(n) = 7/8$, and show the maximum throughput order as a function of the network size. According to Theorem 1 and as can be observed from this figure, the maximum throughput capacity of the network in a grid network with the described characteristics is inversely proportional to the square root of the network size if the probability of each item being in each cache is fixed. Similarly in the random network the maximum throughput is inversely proportional to the logarithm of the network size.

Moreover, comparing scenario i with ii, we observe that the throughput capacity in both cases are almost the same; meaning that using the path discovery scheme will lead to almost the same throughput capacity as the expanding ring discovery. Thus, we can conclude that just by knowing the address of a server containing the required data and forwarding the requests through the shortest path toward that server we can achieve the best performance, and increasing the complexity and control traffic to discover the closest copy of the required content does not add much to the capacity.

On the other hand with a fixed network size, if the probability of an item being in each cache is greater than a threshold ($\Theta(n^{-1/2})$ in cases i, ii and $\Theta(\log^{-1} n)$ in case iii), most of the requests will be served by the caches and not the server, so increasing the probability of an intermediate cache having the content reduces the number of hops needed to forward the content to the customer, and consequently increases the throughput (Figure 2 (b), $n = 10^4$). For content presence probability orders less than these thresholds most of the requests are served by the main server (p_s approaches 1), so the maximum possible number of hops will be traveled by each content to reach the requester and the minimum throughput capacity ($\Theta(\frac{W}{n(1-\rho(n))})$ in cases i, ii, and $\Theta(\frac{W}{\sqrt{n \log n}(1-\rho(n))})$ in case iii) will be achieved.

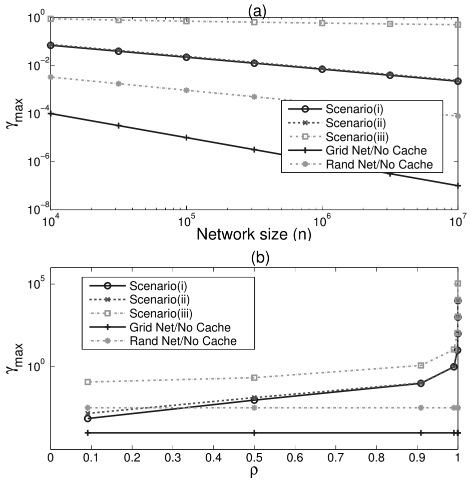

Fig. 2. Maximum download rate (γ_{max}) vs. (a) the number of nodes (n), (b) the content presence probability($\rho(n)$).

Furthermore, if the content availability increases with network growth, higher throughput capacities may be achievable. For example in scenario iii if $\rho(n) = \Theta(\frac{\log n}{\log n + \log \log n})$, then the resulting throughput will be $\gamma_{max} = \Theta(\frac{W}{\log \log n})$ which is much higher than $\Theta(\frac{W}{\log n})$. However, noting that according to Theorem 2, $\rho(n)$ is upper bounded by some values, the achievable capacity will be upper bounded by $\Theta(\frac{W\sqrt{n}\log \log n}{\log n})$ (i) and $\Theta(W\frac{\log \log \log n}{\log \log n})$ (iii).

As may have been expected and according to our results, the obtained throughput is a function of the probability of each content being available in each cache, which in turn is strongly dependent on the network configuration and cache management policy. In the following, we describe this probability in more details and study simple examples in which each item is equally probably available in any cache.

A. Content Distribution in Steady-State

The time diagram of data access process in a cache is illustrated in Figure 3. When a query for content f_i is initiated, the content is available at the requester's cache after a wait time (T_3) which is a function of the distance between the user and the data source (server or an intermediate cache), the data size, and the download speed. An expiration timer will be set upon receiving the data, and this data will be finally dropped after a holding time (T_1) with distribution $\mathfrak{f}_1$ and mean $1/\mu_i$. During this time, the cached data can be shared with the other users if needed. The same user may re-issue a query for that data after some random time (T_2) with distribution $\mathfrak{f}_2$ and mean $1/\lambda_i$. Note that a node will send out a request for a content only if it does not have it in its local cache, otherwise, its request will be served locally and no request is sent to the other nodes. The solid lines in this diagram denote the portions of time that the data is available at local cache.

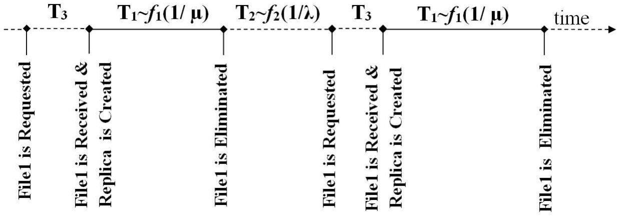

Fig. 3. Data access process time diagram in a cache network

In this work we assume identical content sizes $B_i = B$, and assume all the contents have the same popularity leading to similar request rates $\lambda_i = \lambda$, and the same holding times $\mu_i = \mu$. As the requests for different contents are supposed to be independent and holding times are set for each content independent of the others, we can do the calculations for one single content. If the total number of contents is not a function of the network size, this will not change the capacity order. Suppose that B is much larger than the request size, so we ignore the overhead of the discovery phase in our calculations. Furthermore, if the information sizes are the same and the download rates are also the same, the download time will be a function of the number of hops (h) between the source and the customer; $T_3 = Bh/\gamma$. In the steady-state analysis, we ignore this constant time.

The average portion of time that each node contains a content in its local cache is

$$\rho(n) = \frac{1/\mu}{1/\mu + 1/\lambda} = \frac{\lambda}{\lambda + \mu}, \tag{9}$$

which is the average probability that a node contains the data at steady-state. λ is the rate of requests for a data from each user in case of the data not being available, and μ is the rate of the data being expunged from the cache. Both these parameters are strongly dependent on the total number of users, or the topology and configuration of the network or the cache characteristics like size and replacement policy.

1) Example 1: As a possible example leading to equal probability of all the caches containing a piece of data, which is the basic assumption of Theorems 2 and 3, assume that receiving a data in the local cache of the requesting user sets a time-out timer with exponentially distributed duration with parameter η and no other event will change the timer until it times-out, meaning that $\mu = \eta$. Considering the request process for each content from each user being a Poisson process with rate β, and using the memoryless property of exponential distribution (internal request inter-arrival times), and assuming that the received data is stored only in the end user's cache (the caches on the download path don't store the downloading data), it can be proved that $\lambda = \beta$. Thus we can write the presence probability of each content in each cache as $\rho(n) = \frac{\beta}{\beta+\eta}$.

Figures 4 (a),(b) respectively illustrate the total request rate and the total traffic generated in a fixed size network in scenario i for different request rates when the time-out rate is fixed. The total request rate in the network is the product of the number of requesting nodes and the rate at which each node is sending the request. The total traffic is the product of the total request rate and the number of hops between source and destination and the content size. Small λ means that each node is sending requests with low rate, so fewer caches have the content, and consequently more nodes are sending requests with this low rate. In this case most of the requests are served by the server. The total request rate will increase by increasing the per node request rate. High λ shows that each node is requesting the content with higher rate, so the number of cached content in the network is high, thus fewer nodes are requesting the content with this high rate externally. Here most of the requests are served by the caches. The total request rate then is determined by the content drop rate. So for very large λ, the total request rate is the total number of nodes in the network times the drop rate ($n\mu$) and the total traffic is $n\mu B$. As can be seen there is some request rate at which the traffic reaches its maximum; this happens when there is a balance between the requests served by the server and by the caches, for smaller request rates, most of the requests are served by the server and increasing λ increases the total traffic; for larger λ, on the other hand, most of the requests are served by the caches and increasing the request rate will decrease the distance to the nearest content and decrease total traffic.

Figures 5 (a),(b) respectively illustrate the total request rate and the total traffic generated in a fixed size network in scenario i for different time-out rates when the request rate is fixed. Low $1/\mu$ means high time-out rates or small lifetimes, which means most of the requests are served by the server and caching is not used at all. For large time-out times, all the requests are served by the caches, and the only parameter in determining the total request rate is the time-out rate.

However, when the network grows the traffic in the network will increase and the download rate will decrease. If we assume that the new requests are not issued in the middle of the previous download, the request rate will decrease with network growth. If the holding time of the contents in a cache increases accordingly the total traffic will not change, i.e. if by increasing the network size the requests are issued not as fast as before, and the contents are kept in the caches for longer times, the network will perform similarly.

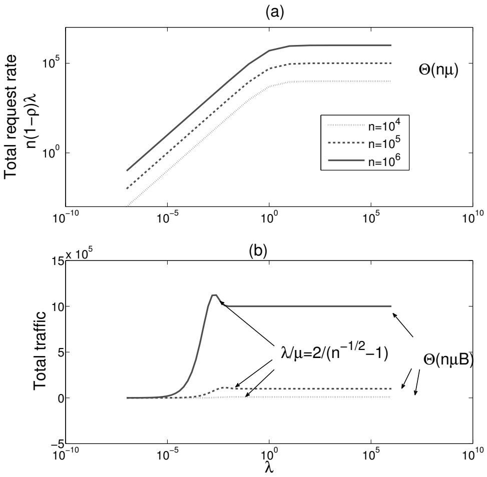

Fig. 4. (a) Total request rate in the network ($\lambda n(1-\rho(n))$), (b) Total traffic in the network ($B\lambda n(1-\rho(n))(p\bar{h} + p_s\bar{h}_s)$) vs. the request rate ($\lambda$) with fixed time-out rate ($\mu = 1$).

2) Example 2: Assume that each cache in level i in a grid network receives requests for a specific document according to a Poisson distribution with rate $\beta'_i(n)$ from all the other nodes, and with rate β from the local user. Note that rate $\beta'_i(n)$ is a function of the individual request rate of users (β) and also the location of the cache in the network. The content discovery mechanism is path-wise discovery, and whenever a copy of the required data is found (in a cache or server), it will be downloaded through the reverse path, and all the nodes on the download path store it in theirs local caches. Moreover, we assume that receiving the data and also any request for the available cached data by a node in level i refreshes a time-out timer with fixed duration D_i. According to [28] this is a good approximation for caches with Least Recently Used (LRU) replacement policy when the cache size and the total number of documents are reasonably large. We will calculate the average probability of the data being in a cache in level i

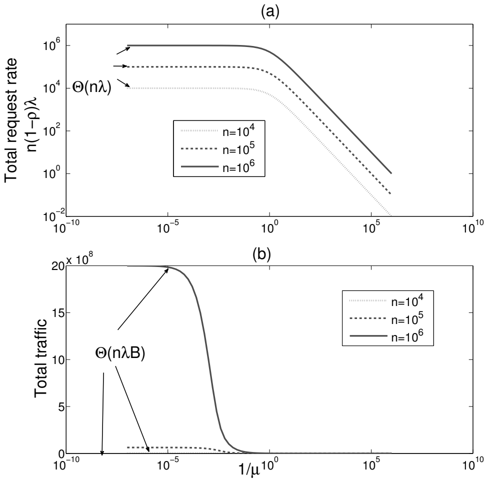

Fig. 5. (a) Total request rate in the network ($\lambda n(1-\rho(n))$), (b) Total traffic in the network ($B\lambda n(1-\rho(n))(p\bar{h}+p_s\bar{h}_s)$) vs. the inverse of the time-out rate ($1/\mu$) with fixed request ratio ($\lambda = 1$).

($\rho_i(n)$) based on these assumptions and then use Theorem 1 to obtain the throughput capacity.

Let random variable $t_{on}(T)$ denote the total time of the data being available in a cache during constant time T. Assume that $N(T)$ requests are received by each node v_i in level i (i hop distance from the server). The data available time between any two successive requests (internal and external) is D_i if the timer set by the first request is expired before the second one comes, or is equal to the time between these two requests. Let τ_i^{req} denote the time between receiving two successive requests. This process has an exponential distribution with parameter $\beta_i = \beta + \beta'_i$. So the total time of data availability in a level i cache is

$$t_{on}(T) = \sum_{k=0}^{N(T)} \min(\tau_i^{req}, D_i), \tag{10}$$

and the average value of this time is

$$\begin{aligned} E[t_{on}(T)] &= \sum_{m=0}^{\infty} E[\sum_{k=0}^{m} \min(\tau_i^{req}, D_i)]Pr(N(T)=m), \\ &= \sum_{m=0}^{\infty} mE[\min(\tau_i^{req}, D_i)]Pr(N(T)=m), \\ &= E[\min(\tau_i^{req}, D_i)]E[N(T)]. \end{aligned} \tag{11}$$

According to the Poisson arrivals of requests with parameter $\beta + \beta'_i$, $E[N(T)] = (\beta + \beta'_i)T$.

$E[\min(\tau_i^{req}, D_i)]$ can be easily calculated and equals to $\frac{1-e^{-D_i(\beta+\beta'_i)}}{\beta+\beta'_i}$. Therefore,

$$E[t_{on}(T)] \;=\; (1 - e^{-D_i(\beta+\beta'_i)})T \tag{12}$$

And finally the probability of an item being available in a level i cache is $\rho_i = \frac{E[t_{on}(T)]}{T} = 1 - e^{-D_i(\beta+\beta'_i(n))}$. Note that $D_0 = \infty$ so that $\rho_0 = 1$.

Now we need to calculate the rate of requests received by each node in level i. We assume that the shortest path from the requester to the server is selected such that all the nodes in level i receive the requests with the same rate. There are $4i$ nodes in level i and $4(i+1)$ nodes in level $i+1$. So the request initiated or forwarded from a node in level $i+1$ will be received by a specific node in level i with probability $\frac{i}{i+1}$ if it is not locally available in that node, so $\beta'_i(n)$ can be expressed as

$$\beta'_i = \frac{(1-\rho_{i+1})(\beta+\beta'_{i+1})(i+1)}{i} \tag{13}$$

Combining equation 13, the relationship between ρ_i and β'_i, and the fact that there is no external request coming to the nodes in the most bottom level ($\beta'_{\sqrt{n}} = \beta$), together with the result of Theorem 1 we can obtain the capacity (γ_{max}) in the grid network with path-wise content discovery and on-path storing scheme which is given by

$$\frac{W\sqrt{n}/4}{\sum_{i=1}^{\sqrt{n}} i \sum_{j=0}^{i} e^{-\sum_{k=j+1}^{i} D_k(\beta+\beta'_k)}(1-e^{-D_j(\beta+\beta'_j)})} \tag{14}$$

Figure 6 (a) illustrates the maximum throughput capacity changes with the network size (n) when $D_i\beta$ is the same for all nodes. It can be seen that the this capacity is inversely proportional to $\sqrt{n}$, just like the throughput capacity when no timer refreshing is available and the downloaded data is stored just in the end user's cache.

Figure 6 (b) shows the capacity versus different values for $D_i\beta$ assuming $n = 10^4$ and same timer expiration time for all nodes. It can be seen that the maximum capacity is very close to $e^{D\beta-1}/\sqrt{n}$. For large $D\beta$ products the probability of the content being available in each and every cache will tend to be one, so all the contents are downloaded from the local cache and no data transfer is needed to be done, therefore the calculated throughput capacity will be very large which means that the all the links are available with their maximum bandwidth.

VI. Conclusion And Future Work

We studied the asymptotic throughput capacity of ICNs with limited lifetime cached data at each node. The grid and random networks are two network models we investigated in this work. The results show that with fixed content presence probability in each cache, the network can have the maximum throughput order of $1/\sqrt{n}$ and $1/\log n$ in cases of grid and random networks, respectively.

Furthermore, since the request rates will decrease as a result of increasing download delays, increasing the content lifetimes according to the network growth may result in higher throughput capacities. However, the throughput capacity is upper limited by some values which comes from the fact that the supportable throughput by each node is limited.

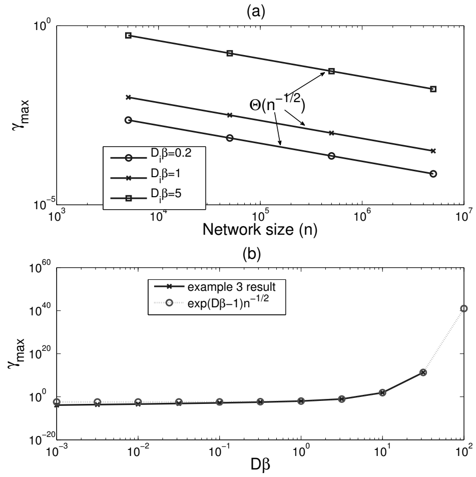

Fig. 6. Maximum throughput capacity (γ_{max}) versus (a) network size (n), (b) Timeout-request rate product (βD).

Moreover, we studied the impact of the content discovery mechanism on the performance. It can be observed that looking for the closest cache containing the content will not have much asymptotic advantage over the simple path-wise discovery. Consequently, downloading the nearest available copy on the path toward the server will have the same performance as downloading from the nearest copy. A practical consequence of this result is that routing may not need to be updated with knowledge of local copies, just getting to the source and finding the content opportunistically will yield the same benefit.

Another interesting finding is that whether all the caches on the download path keep the data or just the end user does it, the maximum throughput capacity scale does not change.

In this work, we have made several assumptions to simplify the analysis. For example, we assumed all the contents have the same characteristics (size, popularity). This assumption should be relaxed in future work. We also assumed that the requester downloads the data completely from one content location. However, if the node that needs the data can download each part of it from different nodes and makes a complete content out of the collected parts, achievable capacities may be different. Proposing a caching and downloading scheme that can improve the capacity order is part of our future work.

References

[1] L. Zhang, D. Estrin, J. Bruke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named data networking (NDN) project," Oct. 2010.

[2] "PURSUIT: Pursuing a pub/sub internet," http://www.fp7-pursuit.eu/, Sep. 2010.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *Communications Magazine, IEEE*, vol. 50, no. 7, July 2012.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *ACM CoNEXT*, 2009, pp. 1–12.

[5] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, and V. Vercellone, "Design considerations for a network of information," in *ACM CoNEXT*, 2008, pp. 1–6.

[6] T. Koponen, M. Chawla, B. G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM*, 2007, pp. 181–192.

[7] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox, "Information-Centric networking: Seeing the forest for the trees," in *HotNets*, 2011.

[8] H. Che, Z. Wang, and Y. Tung, "Analysis and design of hierarchical web caching systems," in *IEEE INFOCOM*, 2001, pp. 1416–1424.

[9] E. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks," in *IEEE INFOCOM*, 2010, pp. 1–9.

[10] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, "On the scale and performance of cooperative Web proxy caching," *SIGOPS Oper. Syst. Rev.*, vol. 33, no. 5, pp. 16–31, Dec. 1999.

[11] E. J. Rosensweig and J. Kurose, "Breadcrumbs: Efficient, Best-Effort content location in cag networks," in *IEEE INFOCOM*, 2009, pp. 2631–2635.

[12] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, no. 1, pp. 77–89, 2005.

[13] M. Tortelli, I. Cianci, L. A. Grieco, G. Boggia, and P. Camarda, "A fairness analysis of content centric networks," Nov. 2011.

[14] C. Westphal, "On maximizing the lifetime of distributed information in ad-hoc networks with individual constraints," in *ACM MobiHoc*, 2005, pp. 26–33.

[15] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, 2000.

[16] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *MobiCom*, 2001, pp. 61–69.

[17] U. Niesen, P. Gupta, and D. Shah, "On capacity scaling in arbitrary wireless networks," *Information Theory, IEEE Transactions on*, vol. 55, no. 9, pp. 3959–3982, 2009.

[18] M. Grossglauser and D. Tse, "Mobility increases the capacity of ad hoc wireless networks," *Networking, IEEE/ACM Transactions On*, vol. 10, no. 4, pp. 477–486, 2002.

[19] J. D. Herdtner and E. K. Chong, "Throughput-storage tradeoff in ad hoc networks," in *IEEE INFOCOM*, 2005, pp. 2536–2542.

[20] G. Alfano, M. Garetto, and E. Leonardi, "Content-centric wireless networks with limited buffers: when mobility hurts," in *IEEE INFOCOM*, 2013.

[21] H. Liu, Y. Zhang, and D. Raychaudhuri, "Performance evaluation of the cache-and-forward (CNF) network for mobile content delivery services," in *ICC Workshop*, 2009, pp. 1–5.

[22] G. Carofiglio, M. Gallo, L. Muscariello, and D. Perino, "Modeling data transfer in content-centric networking," in *IEEE Teletraffic Congress (ITC23)*, 2011, pp. 111–118.

[23] U. Niesen, D. Shah, and G. Wornell, "Caching in wireless networks," *IEEE Transactions on Information Theory*, 2011.

[24] S. Gitzenis, G. S. Paschos, and L. Tassiulas, "Asymptotic laws for content replication and delivery in wireless networks," in *IEEE INFOCOM*, 2012, pp. 531–539.

[25] F. Xue and P. Kumar, *Scaling Laws for Ad Hoc Wireless Networks: an Information Theoretic Approach*. Foundations and Trends in Networking, NOW Publishers, 2006.

[26] M. D. Penrose, "The longest edge of the random minimal spanning tree," *The Annals of Applied Probability*, pp. 340–361, 1997.

[27] M. Raab and A. Steger, "Balls into bins - a simple and tight analysis," in *Proceedings of the Second International Workshop on Randomization and Approximation Techniques in Computer Science*, 1998, pp. 159–170.

[28] H. Che, Y. Tung, and Z. Wang, "Hierarchical Web caching systems: modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, Sep. 2002.