Adaptive Edge-Centric Cloud Content Placement for Responsive Smart Cities [1]

Hassan Sinky*, Bassem Khalfi†, Bechir Hamdaoui†, and Ammar Rayes‡ [1]

* Umm Al-Qura University, Makkah, Saudi Arabia, hhsinky@uqu.edu.sa

† Oregon State University, Corvallis, OR 97331, khalfib,hamdaoui@oregonstate.edu

‡ Cisco Systems, San Jose, CA 95134, rayes@cisco.com

ABSTRACT

In this paper, we propose content-centric, in-network content caching and placement approaches that leverage cooperation among edge cloud devices, content popularity, and GPS trajectory information to improve content delivery speeds, network traffic congestions, cache resource utilization efficiency, and users' quality of experience in highly populated cities. More specifically, our proposed approaches exploit collaborative filtering theory to provide accurate and efficient content popularity predictions to enable proactive in-network caching of Internet contents. We propose a practical content delivery architecture that consists of standalone edge cloud devices to be deployed in the city to cache and process popular Internet contents as it disseminates throughout the network. We also show that our proposed approaches ensure responsive cloud content delivery with minimized service disruption.

I. Introduction

The world has witnessed an unprecedented growth in its urban population throughout the years. Studies show that the world's urban population has expanded by about 60 million per year, and by 2050, 70% of this population is expected to be living in cities[1]. Content delivery networks have emerged as a potential solution for meeting the data demands of cities and the quality of experience (QoE) of users alike. Fog computing [1], [2] has been leveraged to bring data closer to user locations [3] and enhance overall network performance; later evolving to address urban challenges using information, advances in communication technology, and the Internet [4]. Building such an infrastructure is increasingly difficult due to the proliferation of Internet devices and to the huge data demands that these devices would generate. For instance, recent studies show that by 2021, global mobile data traffic will increase sevenfold (from 2016) reaching 49 exabytes per month, most of which will be mobile video content, with a percentage projected to reach up to 78% by 2021 [5].

In traditional IP networks, geographically distributed and limited content delivery nodes service different regions throughout the world [6]. These nodes are typically located at the Internet edges over multiple

[1]This work was supported in part by Cisco Systems and the US National Science Foundation (NSF) under NSF award CNS-1162296.

[1]World population data sheet: http://www.prb.org/

backbones while remotely servicing different regions. Although utilizing caching paradigms to push content closer to the consumer improves network performance [7], users in large urban networks naturally experience added latency due to their increased mobility, congestion, and hops traversed within the network. That is, content in large urban communication networks must traverse the system, to and from a remote data center hosting the content, multiple times resulting in subpar performance. In addition, different users may request the same content incurring excessive duplicate content requests. Thus, pushing content closer to the requesting user using content-centric delivery principles can help improve performance.

Traditional caching methods typically concentrate on one network parameter rather than consider multiple parameters and metrics. This work accounts for multiple network metrics in the caching decision such as latency, distance, caching capacity, city population densities, time variability of content popularity, and intra- and inter-neighborhood content popularity. In addition, we design a complete framework using an existing urban communication network where different caching policies and protocols can be evaluated. Also, we leverage Content-Centric Networking (aka Named-Data Networking) which, unlike traditional IP networking, allows for caching to be analyzed and controlled at much lower level; that is storing small data chunks of the order of a single packet. Traditional methods typically only cache large chunks or full content and are ill-equipped to analyze or control caching granularity. Finally, the large size of caches and massive size of Internet catalogs makes our study a valuable task.

The content-centric networking and delivery paradigm is a proposed Internet architecture that shifts from today's host/IP-centric content delivery principle to a data-centric delivery one, where data is being routed in the Internet based on its content rather than on its physical location [8]. In this paper, we present a framework that leverages edge cloud computing capabilities and content-centric delivery principles to improve content delivery speeds and reduce network traffic in highly populated cities. Specifically, we first propose in Section II a content delivery architecture that consists of standalone edge cloud devices to be deployed in the city to cache and process popular Internet content as it disseminates throughout the network, thereby improving content downloading time, reducing Internet traffic, and avoiding network congestions. For instance, in the case of the LinkNYC network[2], a New York City network we use throughout this work as a use-case for illustrating and evaluating our proposed concepts (see Section II-A for more on LinkNYC), the already-deployed, traditional payphone kiosks, when upgraded with storage and computing capabilities, can play the role of these edge cloud devices, to be referred to as **cloudlets** throughout. Then, in Sections III and IV, we present four complementary content-centric caching techniques that ensure responsive content delivery with minimized service disruption. These techniques exploit: (i) content popularity among city users to make efficient in-network caching decisions to improve content delivery responsiveness; (ii) cooperation and information sharing among city cloudlets to bring Internet content closer to end users with minimum

[2]https://en.wikipedia.org/wiki/LinkNYC

network resource usage; (iii) collaborative filtering theory to provide accurate and efficient content popularity predictions to enable proactive in-network caching of Internet contents; and (iv) user trajectory information obtained through GPS to enable content prefetching, thus providing realtime content access to mobile users with minimized service disruption. We finally conclude the paper in Section V.

II. Cloudlet-Driven Content Delivery Architecture

We present techniques that combine content-centric delivery principles with the edge cloud computing paradigm to improve content delivery responsiveness and reduce network congestions in highly populated cities. Throughout, we use LinkNYC [6] as our use case for illustrating and validating our proposed concepts.

A. The LinkNYC Use Case

LinkNYC [6], an infrastructure project announced in 2014 and became operational in 2016, provides a novel data network offering free gigabit Wi-Fi in New York City (NYC) by replacing thousands of payphones with kiosk-like structures called **Links**, making LinkNYC the largest and fastest free public Wi-Fi network in the world.

Fig. 1 shows the locations of these payphones/Links in each of the 5 NYC boroughs. Each Link is equipped with an 802.11ac Wi-Fi hotspot and basic services such as advertisements, device charging, free voice over IP (VoIP) phone calls, access to city services and maps, etc.

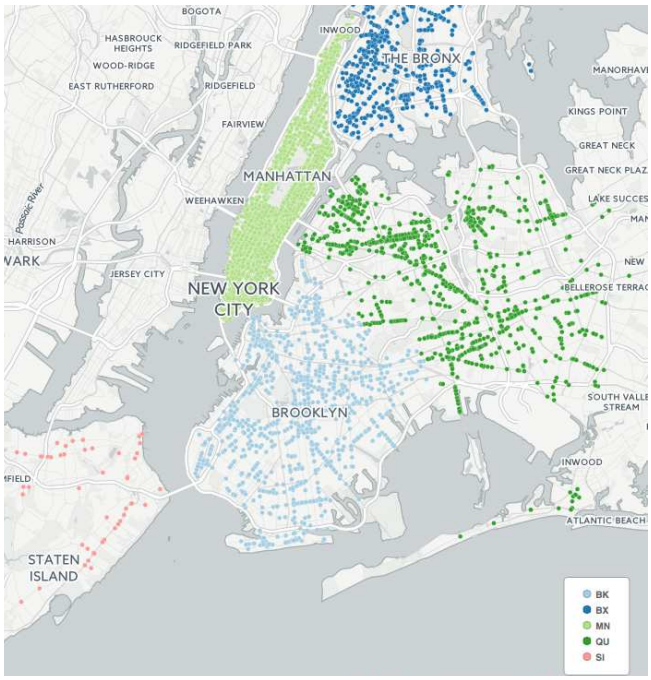


Figure 1. Payphone locations in NYC boroughs: Brooklyn (BK); Bronx (BX); Manhattan (MN); Queens (QU); Staten Island (SI).

We propose to leverage edge cloud computing to improve network responsiveness to content delivery in highly populated cities like NYC. For our LinkNYC use case, this translates into enabling some (or all) of the Links with extra storage and computing capabilities to be able to cache and process content locally, thus reducing the need for having to request content from its source every time a local user requests it. Throughout, we will refer to these capable Links as **content-delivery cloudlets (CDCs)**, which are essentially small-scale cloud datacenters that store data closer to (mobile) end users [9]. Naturally, the number and placement of CDCs depend, among other factors to be

discussed later, on the number and locations of currently installed payphones[3], which are shown in Fig. 1[4] and summarized in Table I for the LinkNYC use case. Manhattan is the most dense of the five boroughs with 3,409 payphones and Staten Island is the most sparse with only 51 payphones. Unlike traditional content delivery networks, where a limited number of remote servers are distributed throughout the world, our proposed approach leverages edge cloud computing to bring content closer to end users by selecting and designating a subset of the LinkNYC's Links to play the role of CDCs. Our preliminary CDC selection and placement approach is presented next.

Table I

PAYPHONES IN THE FIVE BOROUGHS OF NYC

Borough	**# Payphones**	**Avg. distance**	**# CDCs**
Manhattan	3409	43.2 m	50
Queens	1042	136.8 m	25
Brooklyn	1004	150.8 m	25
Bronx	591	125.5 m	20
Staten Island	51	606 m	10
Total	6097	212.5 m	130

B. Cloudlet Selection and Placement

As consumers become increasingly mobile, the placement of CDCs in large cities becomes both crucial and challenging. Specifically, mobile users that undergo frequent handoffs as they move across a path while being connected results in QoE issues [10] if CDCs' placement schemes are not carefully designed. Clearly relying on a single CDC is insufficient to meet the demand of the mobile consumers, and thus, having content readily available in multiple nearby CDCs is indispensable to ensure responsive content delivery and maintain acceptable QoE. In what follows, we discuss clustering approaches that efficiently select and decide on the placement of multiple CDCs to enable content-centric networking and delivery in smart cities, and as done throughout, we consider the LinkNYC network as our use-case for evaluating such approaches.

We apply a hierarchical clustering technique to NYC's boroughs to decide for the placement of CDCs. Since the connectivity of NYC's payphone backhaul is unknown, we assume that Links are physically connected (e.g., by fiber optic cables) to their nearest neighbors. Given a particular NYC borough (e.g. Brooklyn), we construct an Euclidean minimum spanning tree (EMST) using Prim's algorithm [11] where

[3]NYC Open Data: https://nycopendata.socrata.com

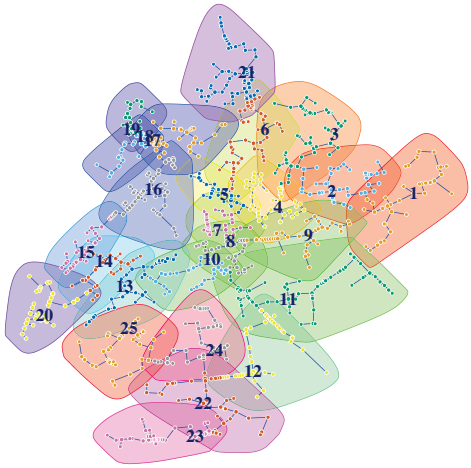


Figure 2. CDC placement: Brooklyn

edge weights are equal to the geographic distances between the Links. The EMST topology for the Brooklyn borough is represented in Fig. 2 by the small dots as the nodes and the lines connecting these dots as the edges. Although we focused here on Brooklyn borough, the same approach applies to each of the other boroughs. Initially, all Links in the borough are considered to be part of the same membership and form a single community. In order to promote and enable cloudlet-driven content delivery, some Links will be chosen to play the role of CDCs based on their average hop counts to the remaining Links within their respective communities. First, the probability that requests are initiated from each Link i is assumed to be proportional to its respective surrounding population density, γ_i, and is defined as $r_i = \frac{\gamma_i}{\sum_{j=1}^{N_l} \gamma_j}$, where N_l is the number of Links in the entire borough network (e.g., Brooklyn). (The population densities of all LinkNYC's Links—γ_i for Link i—are estimated as described in [7]). Let's now denote by $\boldsymbol{S}$ the shortest path matrix that contains the length of the shortest path to and from each Link in the borough network. Given the Link request probability vector $\boldsymbol{r} = (r_1, r_2, \ldots, r_{N_l})$, a weighted average shortest path vector, $\bar{\boldsymbol{s}} = \boldsymbol{S} \cdot \boldsymbol{r}$, is then computed, where each entry value of the vector represents the weighted average hop count to be traversed had all content requested by all borough users been provided through the Link corresponding to the entry. Then, the Link with the minimum sum of weighted average hop counts is selected as a CDC. This ensures that content is placed as close as possible to the geographic location of potential consumers within a community. Once selected, the incident edge between the CDC and the Link with the minimum average hop count is removed, thus forming two disjoint communities. For each of these two communities, a CDC is selected providing the minimum average hop count, resulting in two CDCs and their respective communities. Next, a CDC is selected in the community experiencing the highest average hop count. Once selected the incident edge is removed and the same process repeats until the desired number of CDCs is reached. The CDC placement decision process is a one-time calculation done prior to network deployment resulting in a time complexity of $\mathcal{O}(N_l^3)$. Fig. 3 shows the average hop count for the 5 major boroughs as

the number of CDCs increases.

CDC placement requires an ongoing effort by city network administrators to physically augment cloudlets with additional caching capabilities. That is, increasing the number of CDCs will naturally increase the cost of deployment. Therefore, augmenting *each* cloudlet with caching capabilities is not a practical solution, neither is frequently changing CDC locations to meet demand. That's why our heuristic considers assigning CDCs to only highly populated areas, thereby limiting the incurred cost of deployment. Now the question that arises is how to decide on the appropriate number of CDCs. One approach we use here is to estimate the number of CDCs that corresponds to the 'elbow' in the CDC curve, given in Fig. 3, and use it to be the number of CDCs to be selected. The last column of Table I shows these numbers for each borough, and Fig. 2 shows the clusters/communities for the Brooklyn's network when considering the number of CDCs that corresponds to the elbow value (i.e., # of CDCs $= 25$). The 5 boroughs range in size from small, medium to large, and our framework yields similar performance gains when applied to cities with different sizes.

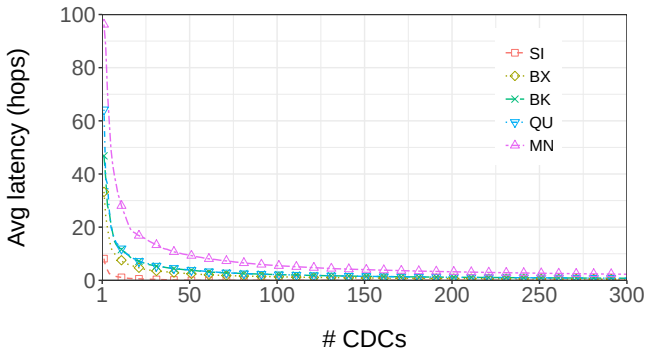


Figure 3. Expected hop count

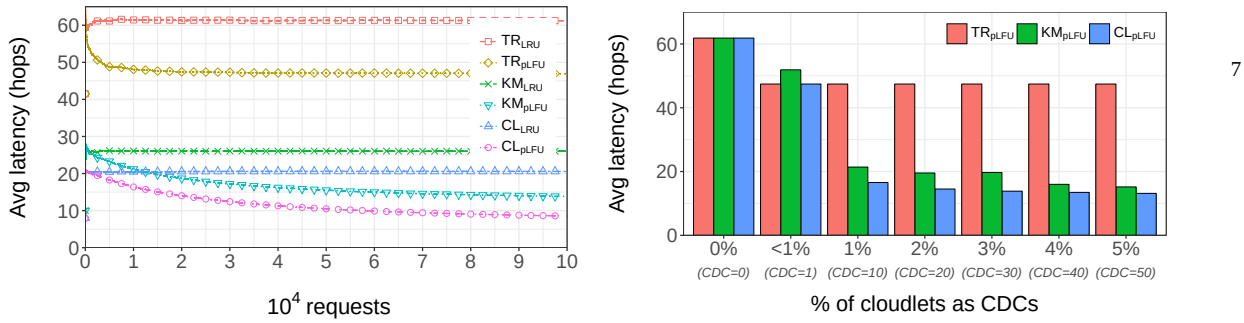
So far, we iterated the benefits of using cloudlet-driven content delivery architectures vis-a-vis of their ability to reduce downloading time and network backhaul traffic. In the following two sections, we present ideas and approaches that enable efficient content delivery in such cloudlet-driven communication architectures.

III. Content-Centric Caching: A Popularity-Driven Approach

Cloudlet-driven architectures do not guarantee the best performance unless key factors like content heterogeneity, user mobility, content popularity, and resource availability are carefully accounted for. Therefore, in the rest of this paper, we focus on content-centric caching approaches that account for these aforementioned factors. Specifically, we will begin in this section by presenting a caching approach that accounts for content popularity information, and discuss in the next section three other content-centric caching approaches that account for other factors. All these proposed approaches are complementary to one another.

A. Popularity-Driven Content Caching

Traditionally, caching consists of fetching content upon request and storing it locally based on some cache replacement policy, such as First-In-First-Out (FIFO), Most Recently Used (MRU), Least Recently Used (LRU), and Least Frequently Used (LFU) [12]. These traditional solutions, however, are not suitable for Internet content delivery in highly populated cities, mainly due to the diversity, volume, and dynamics



(a) Impact of request intensity

(b) Impact of number of CDCs

Figure 4. CDC caching

nature of Internet content. In this section, we investigate a content-centric LFU caching approach that is more suitable for these highly populated cities by incorporating and relying on the popularity of content encountered by the CDCs when deciding on which content to cache. This popularity-based LFU method, denoted **pLFU**, works as follows. Each CDC computes and maintains an estimate of the average number of each content f's requests encountered by the CDC. We propose to estimate this average number periodically using an exponentially weighted moving average approach. Specifically, the average number, $\bar{c}_f^{(k)}$, of encountered content f's requests estimated at the k^{th} update period window is computed as $\bar{c}_f^{(k)} = \beta \bar{c}_f^{(k-1)} + (1-\beta) c_{f,w}^{(k)}$ where $c_{f,w}^{(k)}$ is the number of content f's requests encountered during the k^{th} window period, and β is a weighting design parameter set between 0 and 1. Each content f is then associated with a **popularity index** to be computed by CDC i during the k^{th} window period as $p_{f,i}^{(k)} = \bar{c}_f^{(k)} / \sum_{g \in \mathcal{F}} \bar{c}_g^{(k)}$ where $\mathcal{F}$ is the set of contents encountered by the CDC. Upon arrival of new content and when needed, a CDC uses the popularity index to decide on which content to cache.

B. The LinkNYC Framework

We experimented with the LinkNYC use case to demonstrate the benefits of adopting such a popularity-driven content caching approach in these cloudlet-driven content delivery architectures. Fig. 4(a) shows content delivery latency (measured in terms of number of hops traversed by the content) for the Brooklyn's LinkNYC network when varying the average number of content requests under LRU and $pLFU$. In this experiment, we assume that each CDC is capable of storing about 3% of the total popular content (cache capacity of each CDC is 3% of total content). If requested content is not available at a CDC, it is requested through neighboring CDCs otherwise it must be fetched from the original publisher. The figure depicts the average content delivery latency under: (i) the traditional, single-CDC content delivery approach (TR), (ii) cloudlet-driven content delivery approach with k-means clustering [13] (KM), and (iii) cloudlet-driven content delivery approach with the clustering method described in Section II (CL). For the two clustering methods, the number of CDCs is set to 25 (as determined by the elbow curve shown in Fig. 3).

First, observe that incorporating content popularity when making caching decisions (i.e., pLFU) allows even the single-CDC deployment approach (TR_{pLFU}) to provide a near 25% reduction in average latency com-

pared to using LRU caching (TR_{LRU}). Though reduced, the average latency obtained via the popularity-based content caching approach still remains high (47.5 hops) and is not good enough for dynamically changing environments. Second, the figure also shows that coupling in-network caching through the deployment of multiple CDCs (as in KM and CL) with population-based content caching reduces the latency even further by pushing content of interest even closer to end users. Observe that both KM_{LRU} and CL_{LRU} provide latency reductions of about 58% and 67% compared to TR_{LRU}, whereas KM_{pLFU} and CL_{pLFU} provide about 65% and 80% latency reduction compared to TR_{pLFU}. Note that these latency improvements are a result of the adoption of the cloudlet architecture, which allows to bring and cache content closer to end users. Therefore, such improvements come at the hardware and deployment costs associated with these cloudlets. In the case of LinkNYC, these costs, for instance, should not be significant, since already existing payphone stations and networks have been converted to play the role of cloudlets (though, they still need to be upgraded with extra storage and processing capabilities).

To investigate the impact of the number of CDCs, we show in Fig. 4(b) the latency behavior for Brooklyn's network. As expected, latency improves as more CDCs are deployed. However, it flattens out as the percentage of CDCs increases, and does so around 2 and 3 percent for the Brooklyn network, which corresponds to the optimal number of 25 as determined in Section II and shown in Table I.

IV. Towards Cooperative, Proactive and Predictive Content-Centric Caching: Potential Ideas with Their Associated Challenges

Although accounting for content popularity improves downloading latency, more can still be done. In this section, we propose techniques that rely on (i) cloudlet cooperation, (ii) content popularity prediction, and (iii) GPS trajectory information to improve content delivery responsiveness even further.

A. Cloudlet Cooperation for Faster Content Access

Content caching and placement decisions should depend not only on local but also on neighboring CDC conditions and observations, such as content popularity, storage capacity, content availability in the neighborhood, user population, and link/network condition (congestion, data rates, etc.). Intuitively, when a new content is requested within some local CDC, the decisions on (i) whether to cache the new content or not, (ii) which CDC to cache the content at, and (iii) which existing cache content to evict should involve both local and neighboring CDCs, so that globally optimal placement decisions can be made. For example, if the new content is available at a nearby CDC, then there might not be a need for caching it again at the local CDC, thus saving local cache resources. Now if the new content is not available locally, nor in neighboring CDCs, then the decision whether to cache or not should depend on its community popularity, not just its local popularity. If this content is popular enough to cache, then the decision to where it should

be cached at should weigh in its popularity indexes at the different CDCs within the community. Even if the content is just being requested by a user located within a CDC i, it might be more efficient to cache it at a neighboring CDC if future requests are to be generated by users within the neighboring CDC and/or the neighboring CDC has more available cache space. Designing cooperative content caching and placement approaches that consider the aforementioned performance aspects is an open research problem that has not been addressed yet. And deriving models that capture the various content and network aspects influencing these decisions, such as content popularity, storage availability, content availability, user population, and network condition, is a challenging task that requires further and careful study.

One proposed approach is to introduce a utility function $\mathcal{U}_{f,i}^{(k)}$ that each CDC i maintains for each of its encountered content f, updates every period k, and uses to make content placement and caching decisions.

As an initial step, we propose that this function captures and models the following aspects:

- **Content popularity:** A popularity index of each content as observed by CDC i during update window k. This index is computed by CDC i as explained in Section III-A.
- **Content availability:** A binary availability index of each content, where 1 indicating that the content is cached in CDC i during update window k, and 0 otherwise.
- **Population density:** This reflects the population density of CDC i, as described in Section II-B.
- **Node storage capability:** It captures the storage capacity and availability of CDC i.
- **Network delay:** It represents the delay experienced by a user belonging to one CDC i requesting content cached at a neighboring CDC j. It essentially captures the number of hops, as well as the link bandwidth capacity of each hop, connecting CDCs i and j.

We propose to model the utility as a weighted average of a network cost, ${\mathcal{U}_{net}}_{f,i}^{(k)}$, and a node cost, ${\mathcal{U}_{node}}_{f,i}^{(k)}$; i.e., $\mathcal{U}_{f,i}^{(k)} = \alpha{\mathcal{U}_{net}}_{f,i}^{(k)} + (1-\alpha){\mathcal{U}_{node}}_{f,i}^{(k)}$. The network cost is proportional to the popularity density, the content popularity and availability, and the network delay. It represents a weighted average latency that users' requests generated within all CDCs will experience had content f been cached at CDC i. On the other hand, the node cost is proportional to the local file popularity, local population density and inverse proportional to the intra-CDC delay. Note that this initially proposed node cost function is simple. Other models that capture the nodal cost (processing, storage, memory, energy, etc.) more accurately can be considered. The parameter α balances between the need for having responsive content delivery and that of keeping storage costs low.

As these above network and content conditions change over time, each CDC must periodically maintain and compute utility function values for encountered contents. This could be done by having CDCs query neighboring CDCs for content popularity indexes, population densities, and CDC resource availability, and use this information for updating these values, which are then used as follows for caching and placement decisions. Upon request of a content f, CDC i first finds the least utility value across the set of CDCs that contain content f at period k and checks if i) it is above a target utility value threshold that needs

to be achieved; this can, for example, be the minimum required latency, and $ii)$ $\mathcal{U}_{f,i}^{(k)}$ is less then the least utility value across the set of CDCs that contain content f at period k, then content f is cached at CDC i. Otherwise, no caching takes place.

All these proposed models and approaches need further investigation that we leave for future consideration.

B. Collaborative Filtering for Proactive Content Caching

In the previous section, the focus was on deriving models that capture storage capacity availability, content popularity, user populations, and content availability. In this section, we focus on designing methods that provide effective ways of acquiring the information needed for computing these models. Specifically, we leverage collaborative filtering theory to predict key parameters, such as content popularity, content features, etc., thereby eliminating the need for acquiring it from neighboring CDCs. For instance, content popularity indexes vary over time and across communities, and are unavailable beforehand [14], and hence, it would be very beneficial to have prediction approaches that can provide some accurate estimates in realtime.

One challenge with our popularity-based caching (pLFU) approach (discussed in Section III) is that it may not scale well considering the volume of content that users can be interested in accessing. One approach to overcome this challenge is to use collaborative filtering and low-rank matrix theory [15] to help predict popularity of contents whose popularity indexes are not known yet through the use of contents with known popularity indexes. The idea is that each content can be associated with one (or a combination) of some content categories/interests (e.g., Sports, Politics, Entertainment, etc). For movie content, this categorization could refer to the conventional classifications: action, comedy, drama, documentary, etc. Formally, a content f can be described by a vector of features $\boldsymbol{x}_f$ where each feature measures the degree to which the content falls within a particular category/interest. Now the distribution of content categories within a community served by CDC i during window k can, for instance, be represented by a parameter vector, say $\boldsymbol{\theta}_i^{(k)}$. Note that the distribution of interests may change from one time window to another. The popularity index of content f encountered at CDC i can then be computed as $p_{i,f}^{(k)} = (\boldsymbol{\theta}_i^{(k)})^T \boldsymbol{x}_f$. We can also write popularity indexes of all content using matrix notation as $\boldsymbol{P}^{(k)} = \boldsymbol{\Theta^{(k)}}^T \boldsymbol{X}$ whose rows and columns correspond to the CDCs and contents, respectively. Here, $\boldsymbol{\theta}_i^{(k)}$'s are the columns of matrix $\boldsymbol{\Theta}^{(k)}$ and the $\boldsymbol{x}_f$'s are the columns of the matrix $\boldsymbol{X}$.

We presented some solution approach idea that has great potential for effectively predicting content popularity. This idea requires future investigation.

C. Mobility Prediction for Real-Time Content Delivery

In addition to bringing Internet content closer to end users so that content delivery latency is reduced, cloudlet-driven content delivery architectures in these highly populated cities are required to support and

handle user mobility. In these cities, users will be receiving and downloading content on the move. Therefore, mobile users are expected to experience frequent handoffs across different CDCs during a connection lifetime, resulting in intermittent connectivity and service disruptions, especially near the edge of CDCs' coverage, which is detrimental for mobile service continuity and overall QoE [10]. In this case, even though a handoff may be imminent, interest requests must still be requested, causing potential packet losses and increased response times. Although careful placement of CDCs improves overall network performance, the mobility nature of city users gives rise to service disruption issues that need to be addressed.

For this, we propose to rely on content prefetching approaches for overcoming these issues. Specifically, we rely on GPS technology to predict the mobile user's path and prefetch content on the CDCs located on the user's expected path. For instance, a CDC-aware path P_N, where N is the number of CDCs on a mobile user's path, can consist of a 4-tuple, (C_i, R_i, d_i, s_i), $i = 1, 2, \ldots, N$, with C_i representing CDC i, R_i and d_i representing C_i's throughput and coverage area, and s_i representing the expected speed traveled within C_i's coverage area. Content chunks can then be prefetched on the CDCs belonging to the user's path based on these R_i, d_i and s_i parameters. For example, a mobile user can first obtain a CDC-aware path, $P_N = (C_1, R_1, d_1, s_1), ..., (C_N, R_N, d_N, s_N)$, through a central server where state information and parameters of all CDCs are maintained. The mobile user can also obtain from the server a content-specific manifest file, containing content details such as content size, duration, publisher, etc. Once received, the mobile user parses the manifest file to acquire the content size and in turn the maximum number of chunks based on the maximum transmission unit (MTU) of the network. The mobile user also maintains a current chunk number, S_{cur}, which is used to inform candidate CDCs of the starting chunk number to begin prefetching at as the user moves across a path. Based on the mobile user's current speed, s_i, distance within the CDC's coverage area, d_i, and throughput, R_i, the expected amount to be downloaded within the current CDC can be estimated to $E[D_i] = \frac{d_i}{s_i} R_i$, and hence, the expected chunk number for the candidate CDC to begin prefetching at is $\lfloor \frac{E[D_i]}{MTU} \rfloor + S_{cur}$. This process is repeated until the entire content has been prefetched or the user has arrived at its destination. That is said, what remains for future investigation are a thorough assessment of their ability to prevent service disruption of mobile users, a study of the impact of mobility on the numbers of CDCs that need to be placed, and the finding of the appropriate numbers that account for user mobility.

V. Conclusion

This paper proposes a set of in-network, content-centric caching approaches for large urban cities, and shows that such approaches improve network responsiveness and reduce backhaul traffic congestions. Our presented approaches leverage cooperation and information sharing among network devices, prediction and collaborative filtering of content popularity within city regions, and user trajectory information obtained through GPS to provide faster content delivery, lesser backhaul traffic, and better QoE.

REFERENCES

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[2] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Replisom: Disciplined tiny memory replication for massive IoT devices in lte edge cloud," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 327–338, 2016.

[3] S. Abdelwahab and B. Hamdaoui, "Flocking virtual machines in quest for responsive IoT cloud services," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[4] A. Walid, A. Kobbane, J. Ben-Othman, and M. El Koutbi, "Toward eco-friendly smart mobile devices for smart cities," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 56–61, 2017.

[5] C. V. N. Index, "Global mobile data traffic forecast update, 2016-2021," *Cisco white paper*, 2017.

[6] H. Sinky and B. Hamdaoui, "Cloudlet-aware mobile content delivery in wireless urban communication networks," in *Global Communications Conference (GLOBECOM), 2016 IEEE*. IEEE, 2016, pp. 1–7.

[7] H. Sinky, B. Khalfi, B. Hamdaoui, and A. Rayes, "Responsive content-centric delivery in large urban communication networks: A linknyc use-case," *IEEE Transactions on Wireless Communications*, 2018.

[8] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[9] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, Oct 2009.

[10] H. Sinky, B. Hamdaoui, and M. Guizani, "Handoff-aware cross-layer assisted multi-path TCP for proactive congestion control in mobile heterogeneous wireless networks," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–7.

[11] J. C. Gower and G. Ross, "Minimum spanning trees and single linkage cluster analysis," *Applied statistics*, pp. 54–64, 1969.

[12] H. Al-Zoubi, A. Milenkovic, and M. Milenkovic, "Performance evaluation of cache replacement policies for the SPEC CPU2000 benchmark suite," in *Proc. of the 42nd annual Southeast regional conference*. ACM, 2004, pp. 267–272.

[13] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[14] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1024–1036, 2017.

[15] J. Lee, S. Kim, G. Lebanon, and Y. Singer, "Local low-rank matrix approximation." *ICML (2)*, vol. 28, pp. 82–90, 2013.

Hassan Sinky is presently an Assistant Professor in the College of Computer and Information Systems and Vice Dean of Student Affairs for Student Support at Umm Al-Qura University in Makkah, Saudi Arabia. He received his M.S. and Ph.D. degrees from Oregon State University. His research interests include large urban wireless communication networks, content-delivery and content-centric networks, quality of service and quality of experience methods, cross-layer assisted multi-path TCP and seamless handoffs in wireless mobile scenarios.

Bassem Khalfi (SM'14, M'18) is currently a senior engineer at Qualcomm. He received his Ph.D. in ECE from Oregon State University in 2018. His research focuses on various topics in the area of wireless communication and networks, including dynamic spectrum access and sensing, RF energy harvesting, and content centric networking.

Bechir Hamdaoui (S'02-M'05-SM'12) is a Professor in the School of EECS at Oregon State University. He received M.S. degrees in both ECE (2002) and CS (2004), and the Ph.D. degree in ECE (2005) all from the University of Wisconsin-Madison. His research interests are in the general areas of computer networks and wireless communications. He won several awards, including the ICC 2017 and IWCMC 2017 Best Paper Awards, the 2016 EECS Outstanding Research Award, and the 2009 NSF CAREER Award. He serves/served as an Associate Editor for several journals, including IEEE Transactions on Mobile Computing, IEEE Transactions on Wireless Communications, IEEE Network, and IEEE Transactions on Vehicular Technology. He also chaired/co-chaired many IEEE conference programs/symposia, including the 2017 INFOCOM Demo/Posters program, the 2016 IEEE GLOBECOM Mobile and Wireless Networks symposium, and many others. He served as a Distinguished Lecturer for the IEEE Communication Society for 2016 and 2017. He is a Senior Member of IEEE..

Ammar Rayes (S'85-M'91-SM'15) is a Distinguished Engineer / Senior Director at Cisco Services Chief Technology and Strategy Office working on the Technology Strategy. His research interests include Network Analytics, IoT, Machine Learning and NMS/OSS. He has authored over 100 publications in refereed journals and conferences on advances in software & networking related technologies, 4 Books and over 30 US and International patents. He is the Founding President and board member of the International Society of Service Innovation Professionals www.issip.org, Adjunct Professor at San Jose State University, Editor-in-Chief of Advances of Internet of Things Journal, Editorial Board Member of IEEE Blockchain Newsletter, Transactions on Industrial Networks and Intelligent Systems, Journal of Electronic Research and Application and the European Alliance for Innovation - Industrial Networks and Intelligent Systems. He has served as Associate Editor of ACM Transactions on Internet Technology and Wireless Communications and Mobile Computing Journals, Guest Editor of multiple journals and over half a dozen IEEE Communication or Network Magazine issues, co-chaired the Frontiers in Service Conference and appeared as Keynote speaker at several IEEE and industry Conferences: https://sites.google.com/view/ammarrayes/home At Cisco, Ammar is the founding chair of Cisco Services Research and Cisco Services Patent Council. He received Cisco Chairman's Choice Award for IoT Excellent Innovation & Execution. He received his BS and MS Degrees in EE from the University of Illinois at Urbana and his Ph.D. degree in EE from Washington University in St. Louis, Missouri, where he received the Outstanding Graduate Student Award in Telecommunications.