Search in Digital Pockets: Retrieving Remote Content in Mobile Opportunistic Networks

Suzan Bayhan†, Esa Hyytiä*, Jussi Kangasharju†, and Jörg Ott*‡
† Department of Computer Science, University of Helsinki, Finland, bayhan@hiit.fi, jakangas@cs.helsinki.fi
* COMNET, Aalto University, Finland, esa@netlab.tkk.fi
‡ Technical University of Munich, Germany, ott@in.tum.de

***Abstract*—Due to the fierce competition for the wireless spectrum, operators have recently focused on short-range communications which promises higher spectral efficiency, lower energy consumption, and less strain on the operators' core network. Mobile opportunistic communications, i.e., short-range communications without any network assistance, is of considerable practical value since it entails almost no monetary cost and does not rely on any infrastructure. Compared to the extensive work on opportunistic networks, the predominance of video and other content calls for new content-centric approaches. To this end, the motivation of this paper is to explain how *opportunistic search* can discover the content stored in remote mobile devices and deliver it to the requesting node. We abstract the problem domain as three layers, namely *network topology, content*, and *query*; and describe the interactions among these components. After reviewing each layer, we introduce several schemes for content availability estimation that do not rely on any information exchange but simply use already available information. Additionally, we highlight some open research directions.**

I. Introduction

Mobile devices have been dramatically increasing in number as well as the data volume generated by these devices, e.g. users taking pictures, recording videos. Additionally, ubiquitous sensors produce massive data. These changes have resulted in a shift from the wired Internet being the only data reserve to a system of highly dynamic distributed data sources. Moreover, three facts shift the role of mobile devices from passive consumers to more active entities: (i) device-to-device communications emerge as a solution for removing the strain on the operators who have limited spectrum and resources, (ii) increased concerns for privacy promote the cloud coming closer to the network *edge*, (iii) caching closer to the edge, even on a mobile device, gains more credibility as video traffic accounting for roughly half of mobile data traffic is highly redundant (i.e., skewed popularity distribution). On the other hand, while what is stored on the Internet is well indexed and easily accessible, that is not the case for this newly emerging data reserve. Although users mostly upload the generated content to an Internet service such as Facebook, sometimes publishing on a central server is not desirable nor feasible for various reasons. For these cases, opportunistic networking provides a means of accessing this data reserve.

Mobile devices form an opportunistic network by communicating with nearby devices in a mutual transmission range via their radio interfaces (e.g., WiFi Direct). Typically, end-to-end links are missing in opportunistic networks due to high mobility or low network density. Despite several challenges for viable operation, the motivations for opportunistic networking are manifold: (i) communication is possible even where there is no or weak infrastructure, (ii) robust communications without infrastructure dependency, (iii) direct communication between transmitter and receiver which in turn facilitates higher bit rates, lower delay, and lower power consumption on this direct link, (iv) spectrum reuse gain owing to the lower transmission power, and (v) operators can benefit from mobile data offloading by decreasing the traffic on their core network.

For exploiting the valuable and huge volume of information locally stored in the digital pockets, i.e. mobile devices, efficient mechanisms to discover and retrieve the content are paramount. Use cases may span a wide range of scenarios including search in a dense network, e.g., search for a festival's program during the event, or in a sparse network, e.g., for the shuttle hours to a city in a summer cottage. In the first case, search completion time is more strict compared to the latter. However, owing to the temporal and spatial locality (somebody nearby might have downloaded it already), the search may be completed faster compared to the latter. With advanced mechanisms, e.g., network coding, more bandwidth-hungry content such as video clips can also be delivered in an opportunistic manner.

In this paper, we aim to:

- Clearly describe the *opportunistic search* problem domain, components, and interactions among these components (Section II and Section III),
- Describe a generic framework to understand the impact of content availability, user tolerance to waiting time, and network mobility on search performance based on our previous research [1], [2] (Section IV),
- Introduce several schemes for content availability estimation (Section V),
- Overview the literature (Section VI) and list future research directions (Section VII).

II. Opportunistic Search

As the volume of data in the mobile networks increases in size, efficient access to the relevant information becomes

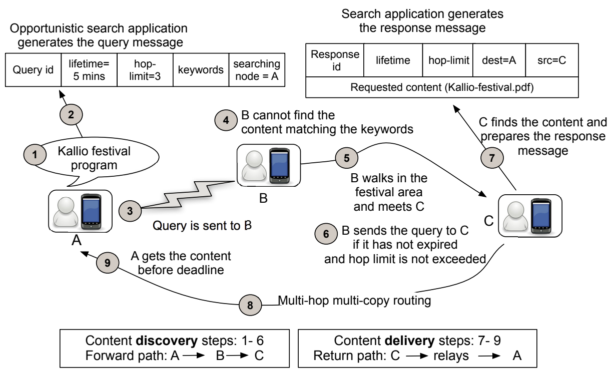


Fig. 1. An example search for the festival program in the festival area.

challenging. First, the relevant content holders should be reached possibly by admitting a search query, and next, data must be retrieved from these nodes. We refer to this process as *opportunistic search*. Requested information may include web pages, video clips, etc. Consider the simple scenario in Fig. 1. *Searching node* A initiates a search query which is carried physically by the mobile nodes and forwarded to other nodes upon contact. Compared to the conventional search, opportunistic search is nontrivial due to two major challenges: lack of a central database to store any content index [3] and time-varying network topology. While nodes may build distributed indices or local knowledge bases to partly alleviate the former challenge, research on opportunistic routing provides some insights on how to handle the latter in protocol design. However, opportunistic search demands special treatment because of the two-step nature of the search.

At the first step, an admitted query is spread in the network with the goal of locating the content provider(s). We call this step *content discovery* and refer to the path a query follows as the *forward or query path*. After the query reaches one of the providers, a response generated by this provider is routed towards the searching node in the second step. We name this step *content delivery* and refer to the path a response follows as the *response or return path*. Content discovery has to tackle the uncertainty of the destination(s) to be reached. In contrast, content delivery is an end-to-end routing. However, it depends on the success of the first step and thereby may not be decoupled from the forward path. Besides, delivering the response may be challenging as the target is a single node rather than a subset of nodes.

Regarding the amount of data bits carried in the query and response, the latter is expected to be (much) larger in size. A typical search query from a mobile device consists of a few keywords [4]. In contrast, considering the predominance of video traffic, response messages carrying the requested content would be much larger. For other queries, e.g., a train schedule, responses may have a comparable size to that of queries.

An ideal search scheme should meet the following goals: (i) it finds relevant content(s) with high probability, (ii) it completes quickly, e.g., before the user loses interest or the content becomes stale, and (iii) it minimizes the number of redundant transmissions to save bandwidth and battery life. However, meeting these goals simultaneously is challenging due to the inherent uncertainty of opportunistic networks. In the next section, we provide a high-level abstraction of the considered system identifying the key components of opportunistic search and interactions among them.

III. Three layers: network topology, content items, and queries

Fig. 2 abstracts the principal components of the considered system: network topology, content items, and queries.

A. Network topology

As Fig. 2 depicts, a node is a mobile device storing content subject to physical constraints (e.g., storage, energy, and wireless radio capabilities) and is coupled with a user who creates content, initiates queries according to her/his interests, and has a social context. Therefore, network topology represents the relations among nodes both in the physical world and the overlaying "social" world. Two nodes are connected at the *wireless connectivity layer* (WCL) if one's transmission is decodable at the other. In the *social connectivity layer* (SCL), two nodes are connected if there is a "social" relation (e.g., friendship, trust) between these nodes [5]. *Social graph* represents existing relations among the nodes in the SCL and remains relatively stable compared to the relations in the WCL.

In opportunistic networks, WCL is subject to frequent changes due to intermittent and highly dynamic connections. Consequently, network connectivity is represented as either the underlying connectivity graph at time t or the expected topology in the long run. While we call the former a *network snapshot*, the latter is known as a *contact graph*. Since human mobility shows some patterns and thereby is highly predictable, contact graphs are considered to be reliable in the long run. On the other hand, these models fall short of modelling the random contacts (e.g., *weak ties*) which are useful for message dissemination.

Node mobility is mostly characterised by the average pair-wise contact rate. As nodes exhibit different social behaviour, node contacts are mostly heterogeneous, i.e., the contact rate between node i and node j is different than the contact rate between node i and node k. The duration of a contact depends on the nature of the contact, e.g., an acquaintance or a close friend, and affects the transmission capacity of the contact. Generally, inter-contact time distribution is approximated by exponential distribution for analytical tractability. However, analysis of real mobility traces suggests that power-law distribution or power-law with an exponential tail represents the mobility characteristics quite accurately [6].

Basic factors determining the network topology are daily habits (e.g., working hours), social interactions among users (e.g., community-based movement), point-of-interests (e.g., café). Knowledge of the network topology is compulsory to develop search algorithms that exploit the transmission opportunities in WCL to the full extent and account for the node behaviour in SCL (e.g., willingness to forward messages).

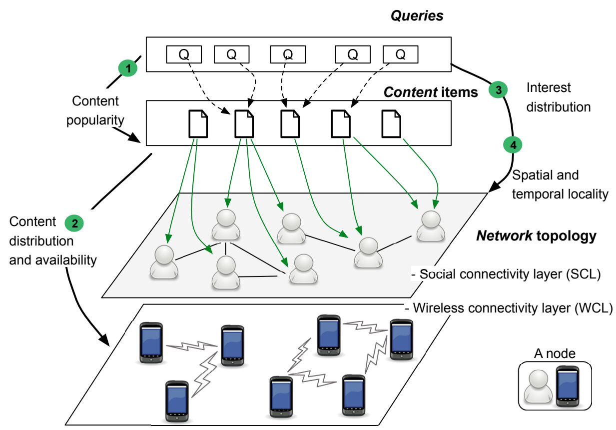


Fig. 2. Three components of the search system: the network topology consisting of social and wireless connectivity sub-layers, content, and queries.

B. Content items

Content items in a mobile device are either generated by the device user, e.g., a video captured at a concert, or downloaded from the Internet and other nodes. For making the local content available to other users, an application running on the user device can provide the necessary means for managing the content (e.g., tagging with keywords or categories) that the user has consent to share or make accessible to others. While there is some literature on how mobile users consume digital data, users' behaviour for generating content (*when*, *where*, and what *type* of content) remains unexplored. Besides, not all consumed content is stored in the device due to the limited storage. Therefore, our knowledge on *content distribution* (which node(s) holds the content) as well as their *availability* (fraction of nodes holding the content) is limited.

Content moves around as the nodes move. To bypass the issue of constantly changing network topology, content can be decoupled from the nodes and be mapped to geographic location [7]. In fact, content-centric design is a perfect match for opportunistic networks as it concentrates on the content rather than targeting a specific host as the provider of this content and initiating an end-to-end routing [8]. For some use cases, content availability could be controlled (e.g., operators can select a certain number of seeds in mobile data offloading scenarios) or tracked by a central entity [9]; for totally self-organized networks, availability may change with time. Although common sense suggests that more popular items will be more available in the network [10], caching policies as well as user behaviour (e.g. deleting the content immediately after accessing) may affect availability.

C. Queries

Research on content-centric networks agrees that owing to the skewed nature of the content request distribution, content delivery algorithms can provide efficient services by exploiting this property. Popular content is stored in the network by popularity-biased caching algorithms while less popular content could be delivered from outside the network, i.e., the origin server. Understanding the distribution of queries, referred to as *content popularity* distribution, is essential to develop caching or replication schemes to achieve high user satisfaction while attaining high resource efficiency.

The search behaviour on high-end phones resembles search on computers more than search on a conventional mobile phone [4]. Hence, it may be reasonable to assume that in opportunistic networks the query to content mapping (dashed arrows in Fig. 2) would follow skewed distribution (e.g., Zipf). Assuming that these submitted queries are fulfilled, i.e., requested content is delivered to the searching nodes, one expects content availability to exhibit a similar distribution to that of content popularity. However, as discussed in Sec. III-B, there may be cases where these two distributions may differ.

Regarding the query to user mapping (arrow 3 in the figure), we expect that users are more likely to search for content that attract their interests. Hence, we call this mapping *interest distribution* which may exhibit spatial and temporal dimension. For example, users are more likely to search for specific content based on their location, e.g., people at the railway station search for a train schedule.

IV. Key factors affecting opportunistic search performance

In this section, we provide some insights on the key factors affecting search success in opportunistic networks. To this end, we consider an elementary model [1], where (i) each node has the same likelihood of storing a particular content item and (ii) each node is equally likely to meet every other node. Moreover, we assume a flooding-based search, which is desirable for opportunistic networks as it requires no state information. We acknowledge that the real world is far more complex (e.g., heterogenous contacts among nodes) and actual scenarios can differ significantly. However, as our aim is to find fundamental factors affecting the search performance that are widely applicable, we assume the parsimonious model with as few parameters as possible that still captures the essential characteristics of opportunistic networks.

More specifically, our model is as follows. In a network of N mobile nodes, searching node n_s admits a search for some content that is held by α fraction of the nodes referred to as *content providers*. Every node receiving a copy of the query forwards it to other nodes it meets. Each content provider receiving the query creates a response and initiates response routing. To alleviate the excessive message replication, the nodes limit the total number of forwarding to M for the forward path and M' for the response path by recording the number of copies in the message header and updating it whenever the message is forwarded, e.g. binary spraying.

Assume that the user is patient in receiving a response for her query: that is, the query does not have strict time restriction. For such a query, we are interested in the probability that the query reaches one of the content providers of the requested content (referred to as *forward path success ratio*) and in the probability that the user receives the requested content (referred to as *search success ratio*). Given that M nodes receive the query, we can approximate the *forward path* *success ratio* as[1]:

$$\text{Forward path success ratio} \approx 1-(1-\alpha)^M. \quad (1)$$

Next, we calculate the search success ratio as:

$$P_s = \sum_{m=1}^{M} Pr\{\text{m content providers are discovered}\} \times Pr\{\text{at least one of m responses reaches } n_s\}.$$

We can expand the above formulation which leads to:

$$P_s = \sum_{m=1}^{M} \binom{M}{m} \alpha^m (1-\alpha)^{M-m} \left(1-(1-\frac{M'}{N-1})^m\right).$$

Let $\gamma = \frac{M'}{N-1}$, i.e., the probability that a response reaches the searching node. Then, we can simplify P_s as:

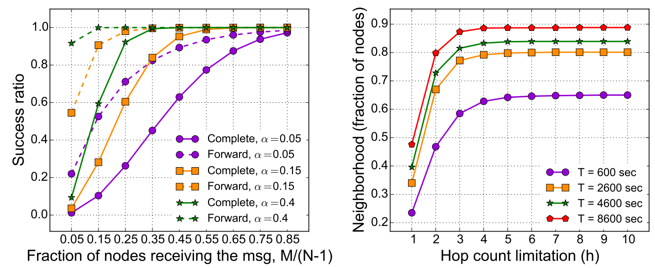
$$P_s = 1-(1-\alpha\gamma)^M. \quad (2)$$

Using (1) and (2), we illustrate the success ratio as a function of allowed message replication, i.e., fraction of nodes receiving the message, under various α in Fig. 3(a). Before discussing Fig.3(a), let us revisit our motivating scenario in Fig. 1. Since A loses interest for the searched information after some time, it may be more practical to set time limitations for the forward (T) and response path (T') instead of M and M'. From a design perspective, replication can be restricted by a hop limit, i.e., a message can be forwarded to maximum h hops. Under time and hop count limitations, an initiated message can reach at most $N_h(T)$ nodes which we refer to as *time-restricted* $h-$*hop neighbourhood* of a node.[2] Hence, in what follows, we replace M and M' in (2) by the related neighbourhood values $N_h(T)$ and $N_h(T')$, respectively.

A. Response paths are more challenging

From Fig. 3(a), we observe that the search messages have to spread to a certain number of nodes for a particular content availability to ensure a given level of search success. The number of nodes receiving the message denoted by $N_h(T)$ depends on node mobility: not only the average inter-contact time but also how diverse the nodes' contacts are. For example, the contact rate may be high, but all contacts are among each other and new nodes are seldom met. Therefore, $N_h(T)$ remains stable over time. Fig. 3(b) plots the cumulative neighbourhood size with increasing h for Infocom06 mobility trace which records the contacts among $N = 98$ nodes.[3] Observation time window and hop limit affect the neighbourhood. Notice the significant increase in $N_h(T)$ at $h = 2$ and the vanishing increase after $h = 4$.

The second parameter α leads us to the following insight. Given that the response path is a search for a single node, we can calculate the success of the response path similar to the success of the forward path where $\alpha = \frac{1}{N}$. This brings us to the fact that under the same T and h constraints, the response path attains a lower success ratio. Generally speaking, response path requires more hops or time for completion than forward path. Note that (2) is derived for a flooding-based scheme making it hard to generalize to more conservative search schemes. Yet, it is instrumental to gain insights on different phases of the search.



(a) Search success, $N = 98$. (b) Infocom06 trace, $N = 98$.

Fig. 3. (a) Search success for various $\alpha = \{0.05, 0.15, 0.40\}$ and (b) cumulative neighborhood for Infocom06 for various T, from [1].

B. Optimal search depth depends on content availability and tolerated search time

To ensure search success to be at least δ_{min}, how should we set the hop count limit? From (2), we can derive the minimum $N_h(T)$ by setting $P_s \geqslant \delta_{min}$. However, it is nontrivial to model $N_h(T)$ as the order of the meetings and T determine the growth in an opportunistic network. Hence, we can rely on numerical results similar to Fig. 3, and obtain the corresponding hop count for a given T and α.

Briefly, content availability, user's tolerance to waiting time for a response, and mobility are three key factors determining the search performance. The product $\lambda_{tot}T$ represents the total number of contacts occurring in the duration of the user's waiting time where λ_{tot} is the aggregate contact rate among all nodes. A search scheme cannot offer much if the network is limited in mobility and density, i.e., low $\lambda_{tot}T$. If both α and $\lambda_{tot}T$ are low representing a scenario where content is scarce and very few contacts are experienced, the search performance is expected to be low. However, if the user does not impose very stringent time requirements or if the content is abundant in the network, search becomes easier. In such cases, it is sufficient to have a low hop count limit (e.g., two or three [1]) to obtain good performance. That is, when $\alpha\lambda_{tot}T$ is large relative to the expected number of content providers met during the search time, limiting the search to a few hops still achieves good results. As $\alpha\lambda_{tot}T$ decreases, the required hop limit to maintain good search performance is larger.

Determining the key factors helps us identify the conditions leading to low performance and engineer the networks such that these conditions are avoided. For example, all search schemes will perform poorly if the user is impatient and the content is only available at a very few nodes. In such a case, we may not be able to increase the user's patience, but we can increase the content availability by caching or active replication. Hence, inference on the operation region is paramount to taking appropriate action.

[1]Please refer to [1] for more details on the derivations.

[2]As forwarding consumes energy, nodes may not always relay the messages. This results in lower $N_h(T)$. However, we assume that all nodes are cooperative, e.g., with the help of some reward mechanisms as in [9].

[3]The traces are available at http://crawdad.org/cambridge/haggle/.

V. Estimating the Content Availability

To set the search parameters appropriately, nodes can infer the content availability *passively* by inspecting the messages they relay or *actively* by exchanging some information to improve their observations. We categorize estimation schemes as *active* and *passive* [2] as illustrated in Fig. 4. In this article, we focus on passive schemes, which can use information carried in queries, responses, or both.

What kind of information does a query and response message carry? Regarding queries, the message header encapsulates the definition of the query (e.g., keywords and searching node) and the route of the query (e.g., the path it has travelled so far). Response headers store the id of the content provider in addition to the query-related information. Assume that nodes run hop-limited routing protocol [1] to search and each node exchanges the status of common messages with each other to avoid the spreading of outdated messages (i.e., already delivered queries and responses). Before deleting outdated messages, nodes update their estimation of the related content's availability. Since a node has multiple observations for the same content, it applies exponential moving average to update estimated availability.

Let $\hat{\alpha}_k(i)$ denote the estimated availability value at node i for content c_k. We calculate $\hat{\alpha}_k(i)$ using four approaches:

- *Exploiting the query hop counts* (**Q-HC**): As proposed in [2], a node can estimate α_k after i nodes have searched in their local storage with a failure of finding c_k by inspecting the number of hops a query has travelled. Estimated availability from a single observation is the inverse of one plus the query message's hop count.
- *Exploiting the response hop counts* (**R-HC**): A node calculates the average path length between a searching node and the content provider. The hop count in a response's header reflects this information as it records the hop count of the received query. Content availability is then the inverse of one plus this path's length.
- *Exploiting the number of carriers* (**Q-NC**): Assume that each node records the number of replications of each query and updates it after forwarding the query to another node. Then, estimated availability is the inverse of one plus the total number of carriers. Note that each node has only local information about the total number of carriers due to the distributed nature of routing.
- *Exploiting the number of content providers* (**R-CP**): A node keeps track of the providers for each content by extracting the provider ids from the response message header. Given that the node is aware of the network size (or estimates it), availability equals to the fraction of nodes with this item.

For each scheme, we calculate the average estimation error over all nodes and all content items. We define the estimation error as the difference between the estimated and actual availability of an item by that node, n_i, using a particular scheme: $\text{Error}_k(i) = \hat{\alpha}_k(i) - \alpha_k$.

We run our experiments using Infocom06 trace.[4] As for content items, we set $M = 100$ and availability is driven

[4]Our analysis on Infocom05 exhibits similar trends, therefore we report only results of Infocom06.

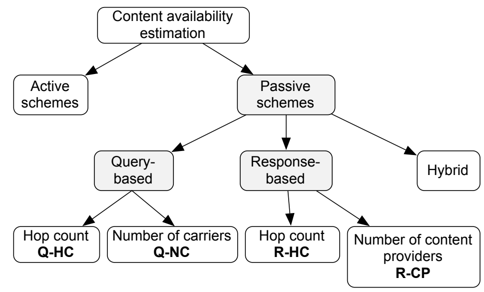


Fig. 4. Availability estimation schemes.

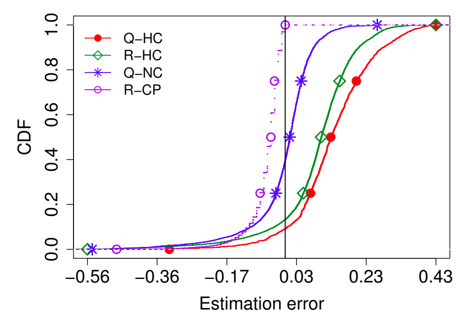
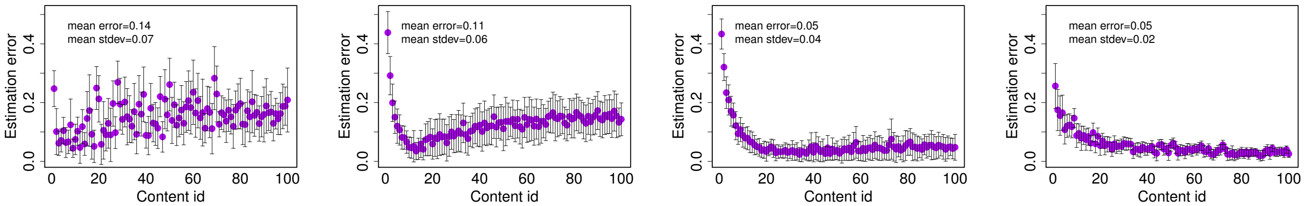


Fig. 5. Distribution of estimation error for Infocom06.

from Zipf distribution with parameter 0.6 while queries are generated according to Weibull popularity distribution with $k = 0.513$ as [11]. Each message's lifetime is two hours, and every minute a random node initiates a query.

Fig. 5 illustrates distribution of estimation errors for all nodes and content items under all schemes. Notice that R-CP only underestimates availabilities as it relies on the number of content providers. In contrast, hop-count-based schemes, i.e. Q-HC and R-HC, result in overestimation. Considering the absolute error (not plotted), our results suggest that Q-NC and R-CP perform very similarly and outperform others in terms of estimation error. Compared to R-CP, Q-NC can be preferable as it does not require network size estimation. Lower accuracy of Q-HC and R-HC is due to the well-known small-world phenomenon. As paths between nodes are short in terms of hop count, nodes tend to overestimate the content availability. For the head part of the content set (i.e., highly available items), estimations by Q-HC and R-HC are usually lower than actual availability, whereas for the tail part estimations are optimistic. Since the considered content availability exhibits long tail property, estimations by hop-count-based schemes deviate drastically from the actual availability.

Fig. 6 plots the absolute estimation error for each content averaged over all nodes' estimation error for this particular content, e.g., nodes exchange their observations. Error bars show the variation among nodes. Note that the content items are sorted in decreasing order of availability. We also calculate the correlation between the content's availability and the related estimation error. The mean error follows the same ordering among schemes: Q-NC and R-CP having the lowest, followed by R-HC and Q-HC. Regarding content availability and related estimation error, Q-HC shows some level of



(a) Q-HC. (b) R-HC. (c) Q-NC. (d) R-CP.

Fig. 6. Average estimation error and standard deviation for each content.

TABLE I
RELATED WORKS: WHETHER AND HOW THEY CONSIDER EACH LAYER.

Related work	Content awareness	Network Topology		Performance Metrics	Hops
		WCL	SCL		
Sermpezis *et al.* [10]	-	✓	✓	Success, delay	1-hop
Fan *et al.* [3]	Geo-community	✓	✓	Success, delay, overhead	2-hop
Hyytiä *et al.* [2]	Estimated availability	✓	-	Search utility, success, overhead	Multi-hop
Bayhan *et al.* [1]	-	✓	-	Success, delay, overhead	Multi-hop
Liu *et al.* [12]	Expertise	✓	✓	Success, delay, overhead	Multi-hop
Pitkänen *et al.* [13]	-	✓	✓	Success, delay, overhead	Multi-hop
Shen *et al.* [7]	Metadata index	✓	-	Average hop distance to content, success, overhead	Multi-hop
Talipov *et al.* [14]	-	✓	✓	Success, overhead, energy consumption	Multi-hop

negative correlation. Q-NC and R-CP exhibit very strong positive correlation, which implies that these schemes are less accurate in estimating availabilities of popular items. Another point to note is that nodes have more observation about popular content items due to skewed popularity distribution and only a few observations about unpopular items. Although we have not observed its impact, Q-NC and R-CP may have lower performance for these items compared to more popular items.

VI. RELATED WORKS

We can classify search schemes as *content-oblivious* [1], [10], [13], [14] and *content-aware* [2], [3], [7], [12] based on their consideration of content characteristics in tuning the search parameters (see Table I). Generally speaking, content-oblivious schemes do not adapt the search strategy according to the requested content, but focus on routing and hitting the content by ensuring a good coverage of the potential content providers. For example, nodes apply hop limitation to each message to avoid the excessive cost of flooding in [1] and authors evaluate the effect of hop count on the attained search success ratio. In [10], the searching node retrieves the content directly from the provider, i.e., one-hop forward and return paths, for which the search delay depends on content availability as well as contact events between the searching node and the providers. The relation between content availability and popularity is modelled as a conditional probability of availability given popularity which reflects both uncorrelated, proportional, and deterministic availability. Unlike [10], a query can take multiple hops in [13] in which each node locally decides to terminate a query based on the expected number of nodes that have acquired the query replica. The authors observe that popular content has a high likelihood of being discovered, but less popular content is found by chance under the examined protocols. The hurdles of lack of awareness in opportunistic networks are mitigated by *redundancy*, i.e., opportunistic protocols replicate messages and route them in parallel rather than routing a single copy and re-transmitting it in case of failures. Some schemes introduce a limited number of replicas at the time of query generation and increase redundancy on the fly at each node [12] whereas some flood the query first and later reduce redundancy by terminating the queries [13], [14]. A query may reach a provider but may have very little time for expiry for content delivery. Since forwarding such a query is probably wasteful, nodes should terminate these queries. The forwarding decision may also consider the distance from the searching node's estimated location to avoid spreading queries with little chance of reaching the searching node [14].

Content-aware schemes [2], [3], [12] guide the search based on the local knowledge about the searched content, e.g., its availability, mapping between content and geolocation, or people's interests. As the content with higher availability is located on average closer to the searching node, a query can be limited to the expected distance from a content provider or each node can decide on terminating the query and sending a response based on the observed availability [2]. In [12], for each query category each node claims its expertise which may not match its real capability for responding to the query. The expertise values are updated based on feedbacks from the query issuer according to the responses.

While the above schemes make no association between the space and the content items/interests, the *geo-community* introduced in [3] represents "clustering of users with common interest and contacting each other frequently at a geo-location". A query is directly mapped to a geo-community. Each node is assessed according to its capability to move to the target geo-community as well as its capability of carrying response back to the searching node. Similar to [3], each content is mapped to some geographic region in [7] where nodes publish the metadata of their content applying a hash function. Multiple nodes located at the corresponding region store the metadata and respond to the requests for this item with the provider's information. While analysis in [7] proves the low communication overhead of the their approach, storage and computation overhead are paramount for mobile nodes.

VII. Open Research Questions

Information-centric networking (ICN) approaches: ICN can mitigate the current issues of opportunistic networks due to unstable connectivity by relaxing the requirement for end-to-end connections and putting the self-contained data units, i.e., content, at the core of protocol design [15]. As argued in [16], MANET scenarios are mostly "data-centric" making content-centricity a natural fit for them. Even if content providers are known by each requester, it may be challenging to reach these destinations using connection-oriented protocols. Moreover, as discussed before, maintaining the information related to coupling between the content and its provider is too troublesome in mobile networks. In ICN, caching and content placement play a key role in providing more efficient services as well as decreasing content delivery cost. However, deciding on *where* (e.g., node or geolocation) to cache as well as *what* to cache turns caching in opportunistic networks into a harder problem [8]. Exploiting the content diversity such as popularity or size is another direction in this line.

Awareness: Nodes can employ passive or active awareness on all layers such as network density, temporal topology characteristics, and content popularity. Estimation schemes under network mobility as well as how the acquired awareness could be incorporated into opportunistic search design are interesting directions to explore.

Quantifying the benefits of indexing: One of the most compelling challenges in search is the lack of the index; nodes do not know the providers. A natural question arises: what would be the performance improvement if each node had an exact copy of the index? Which items to index as well as how to construct and distribute the index are two arising questions.

Measurement-based analysis: Measurements help to comprehend the systems in the real world better. In this line, we think that the network topology layer has received substantial effort whereas the content and query layers remain unexplored.

VIII. Conclusions

Understanding how opportunistic search operates is crucial for retrieving the remote content efficiently and adapting to the content-centric use of the Internet. To this end, we have provided the key components of an opportunistic network with a three-layer abstraction: *network topology*, *content*, and *queries*. In addition to overviewing the characteristics and key challenges related to each layer, we discussed the interactions among layers. Relying on a flooding-based search, we provided some insights on the key parameters affecting search success, namely *content availability, tolerated waiting time*, and *node mobility*. Improving the nodes' understanding of the properties and characteristics of these three layers improves search efficacy. We have introduced simple yet promising content availability estimation techniques. Accurate inference of the content availability is challenging as nodes' observations are fairly limited and dependent on their mobility and content requests. Nevertheless, nodes can observe their states passively while routing the queries and responses.

Acknowledgements

This work was supported by the Academy of Finland in the PDP project (grant no. 260014).

References

[1] S. Bayhan, E. Hyytiä, J. Kangasharju, and J. Ott, "Two hops or more: On the hop limit in opportunistic networks," in *ACM MsWiM*, 2015.

[2] E. Hyytiä, S. Bayhan, J. Ott, and J. Kangasharju, "On search and content availability in opportunistic networks," *Computer Comms.*, vol. 73, pp. 118–31, 2016.

[3] J. Fan, J. Chen, Y. Du, P. Wang, and Y. Sun, "DelQue: A socially aware delegation query scheme in DTNs," *IEEE Trans. on Vehicular Technology*, vol. 60, no. 5, pp. 2181–2193, Jun 2011.

[4] M. Kamvar, M. Kellar, R. Patel, and Y. Xu, "Computers and iphones and mobile phones, oh my!: A logs-based comparison of search users on different devices," in *Int.Conf. on World Wide Web (WWW)*, 2009.

[5] K.-C. Chen, M. Chiang, and H. Poor, "From technological networks to social networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 548–572, Sept. 2013.

[6] T. Karagiannis, J.-Y. Le Boudec, and M. Vojnovic, "Power law and exponential decay of intercontact times between mobile devices," *IEEE Transactions on Mobile Computing*, vol. 9, no. 10, pp. 1377–1390, 2010.

[7] H. Shen, Z. Li, and K. Chen, "A scalable and mobility-resilient data search system for large-scale mobile wireless networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1124–34, 2013.

[8] F. Neves dos Santos, B. Ertl, C. Barakat, T. Spyropoulos, and T. Turletti, "CEDO: Content-centric dissemination algorithm for DTNs," in *ACM MsWiM*, 2013.

[9] X. Bao, Y. Lin, U. Lee, I. Rimac, and R. Choudhury, "DataSpotting: Exploiting naturally clustered mobile devices to offload cellular traffic," in *IEEE INFOCOM*, 2013.

[10] P. Sermpezis and T. Spyropoulos, "Not all content is created equal: Effect of popularity and availability for content-centric opportunistic networking," in *ACM MobiHoc*, 2014.

[11] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in *Int. Workshop on Quality of Service (IWQoS)*, 2008.

[12] Y. Liu, Y. Han, Z. Yang, and H. Wu, "Efficient data query in intermittently-connected mobile ad hoc social networks," *IEEE Trans. on Parallel and Distributed Systems*, 2014.

[13] M. Pitkänen, T. Kärkkäinen, J. Greifenberg, and J. Ott, "Searching for content in mobile DTNs," in *IEEE PerCom*, 2009.

[14] E. Talipov, Y. Chon, and H. Cha, "Content sharing over smartphone-based DTNs," *IEEE Trans. on Mobile Comp.*, vol. 12, no. 3, pp. 581–95, 2013.

[15] M. Amadeo, C. Campolo, A. Molinaro, and G. Ruggeri, "Content-centric wireless networking: A survey," *Comp. Networks*, vol. 72, pp. 1–13, 2014.

[16] M. Varvello, I. Rimac, U. Lee, L. Greenwald, and V. Hilt, "On the design of content-centric MANETs," in *IEEE Intl. Conf. on Wireless On-Demand Network Systems and Services (WONS)*, 2011.