Islam, Hasan M.A.; Chatzopoulos, Dimitris; Lagutin, Dmitrij; Hui, Pan; Yla-Jaaski, Antti

Boosting the Performance of Content Centric Networking Using Delay Tolerant Networking Mechanisms

Published in:
IEEE Access

DOI:
10.1109/ACCESS.2017.2765379

Published: 20/10/2017

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY

Please cite the original version:
ABSTRACT Content-centric networking (CCN) introduces a paradigm shift from a host centric to an information centric communication model for future Internet architectures. It supports the retrieval of a particular content regardless of the physical location of the content. Content caching and content delivery networks are the most popular approaches to deal with the inherent issues of content delivery on the Internet that are caused by its design. Moreover, intermittently connected mobile environments or disruptive networks present a significant challenge to CCN deployment. In this paper, we consider the possibility of using mobile users in improving the efficiency of content delivery. Mobile users are producing a significant fraction of the total Internet traffic, and modern mobile devices have enough storage to cache the downloaded content that may interest other mobile users for a short period too. We present an analytical model of the CCN framework that integrates a delay tolerant networking architecture into the native CCN, and we present large-scale simulation results. Caching on mobile devices can improve the content retrieval time by more than 50%, while the fraction of the requests that are delivered from other mobile devices can be more than 75% in many cases.

INDEX TERMS Communication technology, computer networks, ad hoc networks, heterogeneous network.

I. INTRODUCTION

Today’s Internet architecture relies on the fundamental assumption that there exists an end-to-end path between the source and destination during the communication session. However, the vast majority of Internet usage is dominated by content distribution and retrieval involving a large amount of digital content and this makes the conventional Internet architecture inefficient. In response, Information-Centric Networking (ICN) [1] emerges as a paradigm shift from a host centric to an information centric communication model. It supports the retrieval of a particular content without any reference to the physical location of the content. Named data is the central element of ICN communication instead of its physical location. When a node needs content, it sends a request for a particular content. If any node on the route of the request has the content in its content store, it replies with that content to the request. The main argument for this architectural shift is that named data provide better abstraction than named hosts.

Among all the ICN proposals, Content Centric Networking (CCN) architecture [2] is gaining more and more interest for its architectural design. CCN supports two types of messages: Interest and Data. Each CCN node maintains three data structures; the Content Store (CS), Pending Interest Table (PIT) and Forwarding Information Base (FIB). CCN communication is consumer driven, i.e., a consumer sends Interest packet towards the content source based on the information stored in the FIB. When a node receives an interest, it checks its local cache for the matching content. Otherwise, the node forwards the Interest packet to the interface(s) based on the FIB table until the Interest packet reaches a content source that can satisfy the interest. Intermediate nodes store the interests in the PIT so that the data can be sent back to the proper requester. In addition, PIT is used to suppress the forwarding duplicate interests over the same interface and provides response aggregations. CCN interests that are not satisfied within a reasonable amount of time are retransmitted. As CCN senders are stateless [2], the consumer is responsible for re-expressing interests if not satisfied.

Intermittently connected network topology or network disruption means a significant challenge for ICN deployment.
For instance, name resolution may fail due to network disruptions, especially when the elements of the distributed resolution services are affected by network partitioning. Delay-tolerant networking [3] architectures are proposed for such scenarios, which are characterised by long delay paths, frequent unpredictable disconnections, and network partitions. Such architectures provide flexible and resilient protocols that build an opportunistic network on top of existing underlying Layer 2 and Layer 3 protocols. This is achieved through asynchronous communication along with the use of underlying Convergence Layer Adapters (CLA) (TCP, UDP, Bluetooth, etc.).

DTN is based on store-carry-and-forward models that utilise persistent storage that is distributed in the network. Data are cached in the network and are available for opportunistic transmissions. In particular, content based routing has been explored in DTN architectures [4]. The multitude of the network interfaces in modern mobile devices allows DTN mechanisms to work in parallel with conventional ones. For instance, mobile users who are connected to the internet via the cellular interface, can also use the WiFi-direct interface to exchange messages with their neighbours. DTN architectures assimilate properties of ICN architectures and vice versa.

In this work, we adapt mechanisms from DTN networking to the ICN architecture in order to improve the efficiency of the content retrieval procedure of mobile users. In more detail, we consider the scenario where mobile users request content via a CCN mechanism. These requests can be of many types, such as a single piece of data (e.g., a request for the map of the current location of the user to Google Maps), a data stream (e.g., the homepage of a news website) or related to a specific type of information (e.g., opened restaurants close to the user). Some of the requests can be served more effectively by the cellular network but there are cases, like the third type of the request, that can be served locally. Such requests can potentially be served more effectively by nearby devices or by the cellular towers that have cached the requested content because another user requested that earlier. We considered the case, where contents are cached in the cellular towers but can also be requested from other nearby devices that have stored them. To achieve this, we modified the PIT table of the native CCN while operating on an opportunistic network. The modified PIT table stores the pending requester(s) information in the PIT table. The motivation behind this change is the fact that the original PIT table of the CCN keeps track of the arrival interfaces of the Interest packets in a way that is not feasible in a highly dynamic network.

The contributions of this work can be summarised in the following list:

(i) We explain what are the required modifications in the conventional CCN mechanism in order for it to be functional in a DTN environment.

(ii) We propose a Content Centric DTN network architecture for mobile devices and introduce the required modification for the native CCN so that the native CCN can bridge with the Content Centric DTN protocol. The Content Centric DTN protocol operates independently of existing DTN routing protocols, i.e., DTN routing protocols run on top of the Content Centric DTN protocol. While designing our proposed architecture, we leverage the inherent properties of CCN and DTN architecture [3].

(ii) We discuss the ways via which a mobile user can receive a requested content and show that the download time of a content can be decreased significantly via caching in the cellular access points and in other mobile devices.

(iv) We show that the underlying routing protocol does not have a substantial effect on the download time of contents due to the limited number of hops in the DTN.

Figure 1 depicts the examined scenario where at any time mobile users are connected to the cellular network and are able to potentially communicate directly with other mobile users, depending on the distance between them and the underlying communication framework for the device-to-device communication. All the contents are stored in an origin server, which is located in a cloud infrastructure and can be cached to cellular access points and to mobile devices. Depending on the placement and the number of the cellular access points, the proportion of the content receptions from other mobile users differs significantly, and, as we can see from our large scale simulations (Section V), mobile users are able to successfully handle the requested contents in various different cases of routing schemes. After discussing the related work in the next section, we provide a more detailed explanation of the examined ecosystem in Section III. Next,
in Section IV, we present the proposed protocol which is evaluated in Section V. Finally, in VI we conclude the paper and list our future work.

II. RELATED WORK

Based on the publish/subscribe paradigm, there exist numerous research efforts [5] on device-to-device (D2D) communication in cellular networks, which is defined as direct communication between two mobile users without intervening Base Station (BS) or core network. This concept was first proposed in [6]. Although D2D, from an architectural perspective, seems similar to Mobile Ad-hoc Networks, the key difference between these two is the involvement of the Cellular Access point. Casetti et al. [7] presents content-centric routing in a D2D architecture based on Wifi Direct. The content-centric routing is based on two data structures: PIT of the native CCN and the Content Routing table (CRT). CRT provides the routing information to reach the content items. However, it is not feasible to maintain CRT and the PIT table in dynamic networks where mobile users provide intermittent connectivity. In contrast, our proposed scheme exploits the different PIT table which stores the requester(s) information instead of the arrival face of the original CCN so that the reverse path can be different from the forwarding path of the Interest packets. Nevertheless, most recently, Dabirmoghaddam et al. [8] have concluded that Interest aggregation should not be an integral component of Content-Centric Networks and propose far smaller and more efficient forwarding data structures (e.g., CCN-DART [9]). Another similar effort has been proposed in [10] that allows wireless content dissemination between mobile nodes without relying on infrastructure support. The proposed architecture is based on the publish/subscribe paradigm. Their focus is mainly on implementation aspects based on 802.11 in ad-hoc mode. In contrast, our architecture is based on CCN and DTN architecture and hence there are many architectural differences between their effort and our proposal. Most recently, Liu et al. [11] presents detailed descriptions on content routing based on.ICMANET, and describes a concept model for content routing, and categorizes content routing into proactive, reactive and opportunistic types, then analyzes representative schemes, which can be referred to for the study of joint optimization between content routing and caching in ICMANET. There are also several research efforts in the DTN environment [12], [13]. In [12], the author investigates the possibility of integrating the ICN and the DTN principles into a shared iCDTN architecture. Combining the ICN and the DTN has been demonstrated in a recent effort called RIFE architecture [13]. The RIFE is a universal communication architecture that combines the publish/subscribe based POINT architecture [36] and the DTN through a number of handlers for existing IP-based protocols (e.g., HTTP, CoAP, basic IP) which are mapped onto appropriate named objects within the ICN core. The IP endpoints are connected through the ICN using a gateway. In contrast, our proposed model exploits the DTN architecture in the native CCN architecture that results in a Multihop Cellular Network (MCN) [6]. Amadeo et al. [14] have discussed the potential of the ICN paradigm as a networking solution for connected vehicles. The authors have summarized ICN-VA NETs relevant literature and presented the open challenges in this area. Nevertheless, the analysis of their work shows that the native design principles of ICN well match the main distinctive features of VANETs and the targeted wide set of future vehicular applications. The authors of [15] presents IP-based data DTN routing mechanisms using CCN on the sparsely-connected real vehicular testbed and validate the performance and usability of CCN over VANET. However, their proposed schemes have not considered the forwarding loop and duplicates at the content level while operating on IP-based routing mechanisms. Our proposed model operates independently of DTN routing and can detect the duplicates, and forwarding loop at content level.

User-centric data dissemination in DTNs has been widely explored from various points of view [16]–[19]. Authors of [20] proposed a user-assisted in-network caching scheme, where users who request, download, and keep the content contribute to in-network caching by sharing their downloaded content with other users in the same network domain. Sourlas et al. [16] proposed an information-resilience scheme in the context of Content-Centric Networks (CCN) for the retrieval of content in disruptive, fragmented networks depending on the in-network caching of its attached user. The proposed scheme enhanced the Named Data Networking (NDN) router design as well as the Interest forwarding mechanisms so that users can retrieve cached content when the content origin is not reachable. To achieve this, the authors introduce a new table, called Satisfied Interest Table (SIT), which keeps track of the Data packets that are forwarded to users. In case the content origin is not reachable, the proposed scheme exploits the cache of the other users following SIT entries. However, the proposed scheme performs well only if the users listed in the SIT entries are connected. In [21], the authors present agent-based content retrieval on top of CCN which provides information-centric DTN support as an application module without modifications to CCN message processing. However, their proposed scheme may suffer from PIT bottleneck in delay tolerant environment. In contrast, our proposed scheme exploits the opportunistic communication of mobile users using DTN mechanisms.

From a social-based point of view, the authors of Social-Cast [17] proposed a routing framework that exploits the social ties among users for effective relay selection, while Yoneki et al. [18] discussed the design of a publish-subscribe communication overlay based on the distributed detection of social groups by means of centrality measures. However, this routing mechanisms can be complementary to our proposed scheme, which operates independently of any routing algorithm. Lu et al. at [19] used the K-means clustering algorithm to build the social level forwarding scheme in order to reduce the transmitted messages. This approach raises several inevitable limitations: (i) the interest may fail to reach
the encountered node with the same social level that might have the content to satisfy the interest, (ii) the request from the higher social level will never reach a content provider with a lower social level, (iii) the proposed scheme cannot detect the routing loop of the Interest packet and (iv) the authors have not considered how to optimise similar interests from multiple users. These limitations are addressed in our solution.

D2D communication highly depends on the participation of mobile users in sharing contents. Mobile users may be selfish and would not be willing to forward data to others due to limited resources (e.g., memory, battery power). To handle this issue, a number of incentive mechanisms [22]–[24] has been proposed to motivate users to work in a cooperative way. D2D is still immature and faces many technical challenges and issues regarding aspects such as device discovery, relay selection, security and interference mitigation. The authors of [25] presents an incentive mechanism for data centric message delivery in DTN that exploits the social relationships. This mechanism prevents users from becoming selfish and motivates them to relay the most popular content. Nevertheless, the incentive mechanisms are complementary to our proposed model and can be applied on top of our solution. In this work, we assume that all mobile users are participating in a cooperative way.

III. SYSTEM MODEL

A. PRELIMINARIES

We analyse a CCN architecture where mobile users make a request for named data contents \( c \in C \). We consider a set of mobile users \( \mathcal{M} \) that browse in a large scale metropolitan area and produce their requests for content.\(^1\) A set of Cellular Access Points (CAPs) \( \mathcal{A} \) are deployed in the area and we assume that at any time \( t \) any mobile device \( m \in \mathcal{M} \) is associated with one cellular access point and we denote this by \( m_a(t) \in \mathcal{A} \), while any CAP \( a \) has \( N_a(t) \) mobile devices associated with it at time \( t \).

Each CAP \( a \) also operates as a CCN node by being connected to the fixed network and it maintains a Pending Interest Table \( P_a \) and a Forwarding Information Table \( F_a \). Also, part of its storage \( S_a \) is used for caching contents and works as a Content Store. The cache of each CAP is measured based on the proportion of total contents that it can store \( S_a = \alpha_A |C| \), \( 0 < \alpha_A << 1 \). In addition to the three traditional tables that are used in CCN architectures we add one table, motivated by the work of [16], which stores the satisfied interests. We denote that table with \( D_a \). The entries of \( D_a \) are of the form: \(<\text{content}, \text{user}, \text{time}>\) and work in a similar way to the forwarding interest table, but with the difference that they keep who has satisfied its interest. In addition, we add another table that stores the Pending Requester Information table (PRIT) that stores the requester information instead of the arrival interface. PRIT is used when the CAP receives requests from the DTN interface.

Any mobile node \( m \) is able to communicate directly with its neighbours \( N_m(t) \), whose number depends on the mobility of the users, and the interface used for the connectivity between them.\(^2\) We also denote by \( N_m^k(t) \) the mobile users \( m \) being able to communicate in \( k > 1 \) hops, at time \( t \). Similarly to the CAPs, each mobile node \( m \) keeps three tables a Pending Requester Information Table \( P_m \), a Forwarding Information Table \( F_m \) and a Satisfied Interest Table \( D_m \) and has a Content Store, \( S_m \). The cache of each node is measured based on the proportion of total contents that it can store \( S_m = \alpha_M |C| \). We assume that the storage capabilities of cellular access points is much higher than that of the mobile devices (e.g., some Terabytes compared to a few Megabytes), \( 0 < \alpha_M << \alpha_A << 1 \). Table 1 contains the introduced notation.\(^3\)

B. PROBLEM FORMULATION

Each mobile user \( m \in \mathcal{M} \) requests contents \( c \in C \) at the rate \( r_m^c \). We use the vector \( \mathbf{r} \in \mathcal{R}^{|\mathcal{M}|}_+ \) to denote all the request rates of all the mobile users for content \( c \) and the zero norm\(^4\) of \( \mathbf{r} \), \(|\mathbf{r}|_0\) to indicate the number of the mobile users that are requesting content \( c \). The request rate may depend on multiple factors, but in this work we consider only the popularity of the content \( \pi_c \) and the profile of the user \( u_m \), which indicates the probability of a mobile user requesting each content. We denote the profiles of all users with the vector \( \mathbf{u} \). So the request rate of content \( c \) by user \( m \) is given by:

\[
\begin{align*}
    r_m^c = u_m \cdot \pi_c
\end{align*}
\]

The service rate of an expressed interest from a user \( m \) and a content \( c \) depends on the popularity of the content \( \pi_c \) and the content placement strategy that will be explained in detail in Section IV. An interest in a content from a mobile user can be served in three ways:

1) CORE NETWORK

The mobile device, via the cellular network, sends the Interest packet and the content is retrieved in the traditional CCN way from the Content Store of any intermediate node or from the server of origin, where the content was initially placed upon its creation. At any time \( t \) there exists at least one node that has the required content. In such a case, the service rate of content \( c \) is denoted by \( s_{N1}^c \) and depends on the popularity of the content and the characteristics of the network (load, bandwidth, etc) and the caching policy (e.g., LRU, FIFO, LFU), \( c_{N1} \). Without loss of generality we assume:

\[
\begin{align*}
    s_{N1}^c(t) = c_{N1}(t) \cdot \pi_c
\end{align*}
\]

\(^2\) Bluetooth has a coverage radius of some tens of meters, WiFi-direct of a few hundreds and the soon-to-be-available LTE-direct is expected to have a coverage radius of half a kilometer.

\(^3\) To avoid listing the same variables for both mobile devices and CAPs we use \( x, X \) and \( y \) (i.e. \( x = [m, a] \), \( X = [\mathcal{M}, \mathcal{A}] \) and \( y = \{1, 2, 3\} \).

\(^4\) The zero norm of a vector equals to the non-zero elements of the vector.

\(^1\) The terms user, node and device are used interchangeably depending on the context.
TABLE 1. Notation table.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>set of available contents</td>
</tr>
<tr>
<td>M</td>
<td>set of mobile users</td>
</tr>
<tr>
<td>A</td>
<td>set of cellular access points</td>
</tr>
<tr>
<td>(\alpha_N)</td>
<td>cache capacity of mobile users (M) or CAPs (A)</td>
</tr>
<tr>
<td>(N_c(t))</td>
<td>mobile users accessible by user (m) or CAPs (A) at time (t)</td>
</tr>
<tr>
<td>(P_x)</td>
<td>Pending Interest Table of user (m) or CAP (A)</td>
</tr>
<tr>
<td>(F_x)</td>
<td>Forwarding Information Table of user (m) or CAP (A)</td>
</tr>
<tr>
<td>(S_x)</td>
<td>Content storage of user (m) or CAP (A)</td>
</tr>
<tr>
<td>(\pi_c)</td>
<td>Popularity of content (c)</td>
</tr>
<tr>
<td>(u_m)</td>
<td>Content request profile of mobile user (m)</td>
</tr>
<tr>
<td>(r_m^c)</td>
<td>Request rate of content (c) from mobile user (m)</td>
</tr>
<tr>
<td>(s_N^c(t))</td>
<td>Service rate of content (c) at time (t) through network (N)</td>
</tr>
</tbody>
</table>

2) CELLULAR ACCESS POINT

The mobile device downloads the cached content from the cellular tower because another user had requested the content earlier. Given that the available cache of each cellular tower is limited compared to the storage size of the server of origin, the cached contents are limited (\(\alpha_A|C|\)) but, depending on the caching policy, can achieve a high hit rate due to the popularity distribution of the contents and the spatial skewness [26]. In that case, we denote the service rate with:

\[ s_{N_2}^c(t) = \alpha_A \cdot c_{N_2}(t) \cdot \pi_c \]  

(3)

3) DELAY TOLERANT NETWORK

The mobile device gets the content from another mobile device via a single-hop or a multi-hop path. The number of the hops depends on (i) the physical distance between the users, (ii) the number of the users and (iii) the popularity of the content. Popular contents are more probably found closer to the user who initiated the request. Although mobile devices are not able to cache many content items, the social relationship between mobile users that have, with high probability, similar mobility patterns, makes it probable for two socially close mobile users to express interest in similar items [27]. In that case, we denote the service rate with:

\[ s_{N_3}^c(t) = \alpha_M \cdot c_{N_3}(t) \cdot \pi_c \cdot \prod_{m \in N_m(t)} u_m \]  

(4)

We employ a Markov process \(X(t), 0 \leq t < \infty\) with stationary transition probabilities that shows the number of the nodes in the whole ecosystem (mobile users, cellular access points, the server of origin as well as the network components such as switches that are part of the CCN ecosystem that have the content \(c\) in their caches). If at any time \(t\), \(X(t) = 0\), this will mean that the content is not available at all, which can be true only in the case of a very unpopular item that is not cached in any node and the server of origin is not accessible because of network partitioning. However, although this is not realistic, we can use the Markov process as a birth-death process with a single absorbing state, which we define to be \(X_c(t) = 0\) in order to then use the absorption time formula [28] that includes the cost parameters for each type of network as an objective function to optimise. In more detail [28]:

\[ T_n^c = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} + \sum_{i=1}^{\infty} \frac{\rho_i}{\lambda_i} \sum_{j=1}^{\infty} \frac{1}{\lambda_j} \]  

(5)

\[ \text{if } \sum_{i=1}^{\infty} \frac{1}{\lambda_i} \rho_i < \infty \]  

(6)

and \(T_n^c = \infty\), if \(\sum_{i=1}^{\infty} \frac{1}{\lambda_i} \rho_i = \infty\), where: \(\lambda_i^c\) is the birth rate of the process at state \(n\), \(\mu_i^c\) is the death rate and

\[ \rho_i^c = \prod_{i=1}^{n} \mu_i^c \]  

(7)

The birth rate of the process at state \(n\) and for content \(c\), \(\lambda_n^c\), depends on the request rates for the examined item of each of the users \(r_n^c\).

\[ \lambda_n^c \sim \sum_{m \in M \cap \|r^c\| = n} r^c_m \]  

(8)

while the death rate depends on the type of the service rate and the caching policies. The required time for the Markov process to reach the absorption state depends on the initial state and the difference between the service rates and the request rates.

The service rate depends on the probability of a content being placed close to the mobile users that generate requests for it. The probability of a content \(c\) being cached in the CAP which mobile user \(m\) is associated with at time \(t\), \(m(t)\) is:

\[ p_m^c(t) := P[c \in S_m(t)] \]  

(9)

and the probability of \(c\) being stored in at least one of \(m\)’s neighbours is

\[ p_{N_m}^c(t) := 1 - \sum_{j \in N_m(t)} P[c \notin S_j(t)], \]  

(10)

while for \(K\) hops away from \(m\), the probability of \(c\) being cached is:

\[ p_{N_m^K}^c(t) := 1 - \sum_{j \in N_m^K(t)} P[c \notin S_j(t)]. \]  

(11)

So the probability for a mobile user not being able to retrieve \(c\) from the access point that he or she is associated with and from any mobile user whose distance is at most \(K\)-hops is:

\[ p_m^K(C, t) := 1 - p_m^c(t) - p_{N_m}^c(t) - \sum_{k=2}^{K} p_{N_m^K}^c(t). \]  

(12)

\(^5\) Small values of \(K\) are enough for successful content discovery [29].
The cellular access points connect the CCN with the static links between the nodes to the highly dynamic and unpredictable DTN.

We also define the probability of a content \( c \) being cached in the cellular access point \( a \) of at least one of the mobile devices that are associated with \( a \):

\[
p^a_t[N_a(t), t] = 1 - P[c \notin S_a(t)] \cdot \prod_{j \in N_a(t)} P[c \notin S_j(t)] \quad (13)
\]

The size of the Content storage in the CAPs and mobile devices, and more specifically the proportion of the total items they can store is what affects \( p^a_t[N_a(t), t] \) and \( p^c_t[N_a(t), t] \). Another determinant parameter is the number of mobile users that are associated with the same access point as the user that requested a content item and, consequently, the diversity in that are associated with the same access point as the user that requested a content item and, consequently, the diversity in the subset of the objects that are cached in all these devices. We denote with \( \mathcal{C}_a(t) \subset \mathcal{C} \) the set of the content items that are cached in at least one device that is accessible from \( CAP \) or are cached in \( a \). Then equation (13) can be expressed shortly as \( P[c \in \mathcal{C}_a(t)] \).

Next, in Section IV, we present a protocol that determines which contents should be cached in each device and for how long. The protocol is designed to consider highly dynamic mobile users with limited resources as well as the static access points that operate as the glue between the dynamic users and the fixed infrastructure.

### IV. PROTOCOL

The original design of CCN is based on the fact that the multiple network interfaces can be integrated via the mechanism of the forwarding information base [2]. Each entry on the FIB points to a list of interfaces that can be used to forward Interest packets towards the desired content producer. At this point, the traditional CCN can be combined with DTN network protocols, as presented in figure 2. The integration of DTN architecture with the native CCN architecture results in a Multihop Cellular Network (MCN) [6]. The general concept of MCN comprises a cellular network in which user devices can communicate with each other, either via means of a conventional cellular mode or via means of direct D2D communication if they are mutually reachable. To enable this paradigm, the functionalities of the proposed protocol can be decomposed into three parts:

1) **The Control Plane**: that performs packet (Interest/Data) management. The control plane is implemented on top of the DTN mechanisms, and its functionalities are responsible for performing specific actions based on the packet type (Interest/Data). To achieve this, the control plane inserts the meta-information in the DTN messages.

2) **The Forwarding Plane**: that consists of two parts and, depending on the type of the node, it can be either the native CCN forwarding or the DTN forwarding (Store-carry-and-forward). This module provides an interface between the CAP and the mobile nodes so that the CAP can hand over the packet to a mobile node. The mobile node exploits DTN architecture to forward the packets in D2D fashion while operating in an opportunistic network without the intervention of the cellular network. The CAP includes a separate PIT table called the Pending Requester Information Table (PRIT) which stores the requester(s) information instead of the arrival faces of the Interest packets. The mobile nodes only use our proposed architecture while operating in a DTN environment, i.e., the control plane is implemented on top of the DTN forwarding plane and enables the host centric DTN to perform content centric fashion. To bridge between CCN and DTN, each message carries meta-information of the CCN mechanism that assists the content centric operation in DTN environment.

3) **The Routing Decision Engine**: is the process by which one router sends packets to another router by means of routing protocols which decide the appropriate path for the packet. The routing protocol assists the router in choosing the best path out of many paths. The routing decision engine operates independently on top of our proposed model.

The proposed protocol deals with two control decisions:

1) **Request/Response Processing**: Although the CAPs operate as conventional CCN nodes regarding the forwarding and the routing of Interest and Response packets, it is not the same for mobile users. Whenever a mobile user of a CAP receives a content request or a content response, there is the question of what actions should be taken?

2) **Content Management**: Given that a mobile user or a CAP has a content item, should it store it in the content store or drop it? The CAPs have higher storage capabilities than the mobile users, but still they can not cache all the available contents.

### A. REQUEST PROCESSING

In the relatively static CCNs, the Interest packets are propagated as upstream towards the potential data sources, while leaving a trail of *bread crumbs* for the matching data packets to follow back to the original requester(s). On the other hand, in dynamic environments the nodes are mobile and the connections are intermittent, which means that it is not feasible to keep track of the changes in the network topology. Unlike the conventional PITs in CCN, mobile users keep the address information of the requester(s) in the Pending Requester Information Table (PRIT) so...
that they can forward similar content towards potential requester(s). PRIT is also used to detect forwarding loop and aggregate the similar interests. Mobile users exploit the Satisfied Request Information Table (SRIT) to remember all the satisfied interests of the requester(s) so that it can provide information on the potential content source for the similar interests in future. By doing that, an intermediate node having an entry matching with the interest packet in the SRIT can forward the Interest packet towards those potential content provider(s). The CAP acts similarly to a mobile node if it receives the Interest packet from the DTN interface. Nevertheless, if the CAP has FIB entry for this Interest, it can also apply the native CCN mechanism. The overview of the request processing is presented in Algorithm 1.

On the reception of an Interest packet, a mobile node initially searches in its Content Store and if there is no match, the node checks its SRIT table to verify if there is any entry matching with the Interest packet. If any matching is found, the node forwards the request towards those potential content provider(s) from SRIT. The node also enters the Interest packet in the PRIT table. The PRIT is used to keep track of the IDs of the interest(s) creators that are used as destinations in the response packets. In more detail, upon the reception of an Interest packet the mobile node checks its PRIT. If there is an older entry for the same content, it updates the entry only if the requester is different, otherwise it drops the Interest packet. On the other hand, the CAP first applies the native CCN mechanism, i.e., it searches the content store to verify if it can satisfy the request. If there is a match, the CAP sends the content back to the requester. If no matching is found, the CAP forwards the request further, based on the information of the FIB. The CAP can also forward the request to the mobile node which runs our proposed architecture. Before forwarding the request to the mobile node, the CAP will store the requester information in the PRIT table, but only if it receives the request from the DTN interface.

Regardless of the total number of users, our proposal does not spread the Interest packets all over the ecosystem because it is inefficient and not worthwhile doing since the mobile nodes are submitting their requests in parallel to both the CAPs they are connected to and their neighbouring mobile devices. More importantly, the respective CAPs inform the mobile nodes whether there exists another mobile node that has the requested content in the same cell, and depending on the level of the assistance from the CAPs, as will be discussed in the next section, the mobile nodes can either receive their request via a multi-hop-but-short path from another node in the same cell, or via a two hop path with the help of the CAP. So, a request as shown in Figure 3 can be served in four ways: (A) from the Content Store of the associated CAP, (B) via the associated CAP that retrieved the content from the conventional CCN network, (C) from another mobile node that sent the content via a multi-path among the other mobile nodes, and (D) from another mobile node that sent the

Algorithm 1 Processing Interest Message

```plaintext
1: key ← [Interest]
2: if key in Local Cache then
3:    content ← Cache(key)
4:    end if
5: if content ≠ NULL then
6:    response ← createResponse(content)
7:    if current_node = mobile user then
8:      requester ← [Interest]
9:      Send response to requester following PRIT
10: else
11:    Send response following PIT breadcrumb
12: end if
13: else
14: if current_node = mobile user then
15: satisfied_req_provider ← lookup_SRIT(Interest)
16: if satisfied_req_provider ≠ NULL then
17:    Send Interest to satisfied_req_provider
18: else
19: pending_requester ← lookup_PRIT(Interest)
20: if requester ∈ pending_requester then
21:    drop the interest packet
22: else
23:    Add requester to PRIT table
24:    forward the Interest to next Hop
25: end if
26: end if
27: end if
28: if current_node = CAP then
29: FIB_entry ← native_CCN_mechanism(Interest)
30: satisfied_req_provider ← lookup_SRIT(Interest)
31: if FIB_entry = NULL then
32: if satisfied_req_provider ≠ NULL then
33:    Send Interest to satisfied_req_provider
34: else
35: pending_requester ← lookup_PRIT(Interest)
36: if requester ∈ pending_requester then
37:    drop the interest packet
38: else
39:    Add requester to PRIT table
40:    forward the Interest to mobile user
41: end if
42: end if
43: end if
44: end if
45: end if
```

content to the CAP, which then forwarded the content to the requester.

B. RESPONSE PROCESSING

Algorithm 2 presents an overview of the response processing on a network node. When the Interest packet reaches a node having content matching with the Interest packet, the node

6Without loss of generality we assume that the id of user m is m.
Algorithm 2 Processing Response Packet

1: if current_node is mobile user then
2: .destination_id ← [Response]
3: .content_provider ← [Response]
4:  insert content_provider in SRIT table
5: end if
6: if current_node is the destination then
7:  notify application
8:  key ← [Response]
9:  pending_requester ← lookup_PRIT(key)
10:  if pending_requester is empty then
11:     drop the packet
12:     return
13:  else
14:     forward response to pending_requester
15:     return
16: end if
17: end if
18: if current_node is CAP then
19:  if response is received from DTN interface then
20:     key ← [Response]
21:     content_provider ← [Response]
22:     insert content_provider in SRIT table
23:     pending_requester ← lookup_PRIT(key)
24:     if pending_requester is empty then
25:         drop the packet
26:     else
27:         forward response to pending_requester
28:     end if
29:     else
30:         follow the native CCN mechanism
31: end if
32: end if

constructs a response packet with the content and sends it back to the originator of the request. If the intermediate node is a mobile node, the node checks the PRIT table and removes the entry if there is a match for the response packet. If the PRIT entry has the information on multiple requesters, the intermediate node adds all the source IDs of those requesters to the response packet as meta-information. If the intermediate node does not find any matches in the PRIT table, it simply forwards the response packet to the next best contact. Subsequently, if the response packet reaches the target node, it checks the meta-information to verify if there is any other pending requester(s) who requested this content. If there exists no pending requester information, the recipient node drops the packet to avoid further transmission by the DTN mechanism. Otherwise, if the node finds other pending requesters, it will forward the response to those pending requesters. If the meta information has multiple pending requesters, the node adds one requester as the destination address for the response and other requester(s) as meta-information. If the intermediate node is the CAP, it checks both PIT and PRIT to forward the response in an appropriate manner. If the CAP finds a match in its PIT, it follows the native CCN mechanism. A match in PRIT follows our proposed scheme.

C. CONTENT MANAGEMENT

Mobile nodes can provide storage memory depending on their resource availabilities and policies. Using its storage memory, a mobile node can serve as the network medium to share the content. Furthermore, this cache can also be used by store-carry-and-forward based DTN protocols. However, the persistent storage of the DTN protocol keeps the message until the successful delivery of the message to the next best opportunistic contact. In our proposed architecture, the storage memory of mobile node keeps the response packet to satisfy the future requests. However, only in the ideal case is the mobile node’s storage big enough to store all received content. Under the assumption that a mobile node can only store a small proportion of the total contents, a caching policy is required.

Additional information, such as the popularity distribution of the contents and the request profile of the mobile users, can be used by a caching policy to determine the probability of each content being requested by the user and, based on that, a decision is made whether a newly received content be dropped or replaced with the unpopular one, given that there is no available space in the mobile node. However, the CAP has information about the stored content in the whole cell that is not utilised. We utilise this additional information by defining the expected retrieval cost of each content by combining this information: (i) the popularity of each content item, (ii) the profile of the users and the (i) estimated time required to retrieve the content via one of the aforementioned ways. This costs are calculated with the assistance of the CAPs, which is able to recommend a mobile node on whether to keep an item or not.
V. EVALUATION

We evaluate our proposed architecture using the Opportunistic Network Simulator (ONE) [30]. The goal of our evaluation is to investigate the performance of our proposal in terms of (i) Average end-to-end delay, (ii) service load ratio, (iii) packet drop ratio and (iv) Traffic split. Table 2 contains a description of each of these metrics.

The ONE simulator contains map data of the Helsinki downtown area (e.g., roads, tram routes and pedestrian walkways) and various Map-based Movement models: (1) Random Map-Based Movement, (2) Shortest Path Map-Based Movement, and (3) Routed Map-Based Movement. We employ the Shortest Path Map-Based Movement since it is more realistic because the mobile users, after choosing a destination point on the map, follow the shortest path to that point from their current location. The destination point is chosen randomly from a list of Points of Interest (POI), which includes popular real world destinations (e.g., shops, restaurants, tourist attractions). The simulation area approximately is 20km².

In the simulation, we considered mobile users that are either walking at a speed that is in the range of 1.8 kilometres per hour to 5.4 kilometres per hour or driving a car or using the tram. We categorised the mobile users into two groups: (i) requesters and (ii) intermediate users. The requesters were 10 and the intermediate users were 150. All of them were divided into four different groups and assigned with different probabilities of choosing the next group specific POI or random places to visit. Regarding the content generation, we considered content generated by 10 other mobile users or from non-mobile content generators (e.g., a news website). Apart from the mobile users, we also considered 30 CAPs that have caching capabilities.

None of the users had any content in the beginning of the simulation, but whenever one requester imposed a request on a CAP, the content was retrieved from the content provider in the cloud if it had not already been cached from a previous request, and delivered to the requester. The simulation time was 5 days and we used the first day as a warm up phase. All the details of the simulation parameters are listed in Table 3.

### TABLE 2. Performance metrics.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average end-to-end delay</td>
<td>The average time passed to receive a content in response to a request.</td>
</tr>
<tr>
<td>Packet drop</td>
<td>The amount of packets (Interest/Data) that is effectively suppressed by the content router.</td>
</tr>
<tr>
<td>Traffic split</td>
<td>The service rate, i.e., the amount of requests processed by different types of nodes in the network: mobile user, CAP, content source.</td>
</tr>
<tr>
<td>Service load</td>
<td>The amount of requests processed by the content provider.</td>
</tr>
</tbody>
</table>

### TABLE 3. Simulation parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Duration</td>
<td>5 days (432000 s)</td>
</tr>
<tr>
<td>Number of Requesters</td>
<td>10</td>
</tr>
<tr>
<td>Time interval of generating Interests</td>
<td>5 min</td>
</tr>
<tr>
<td>Number of Relay Nodes</td>
<td>160</td>
</tr>
<tr>
<td>Number of Access Points</td>
<td>30</td>
</tr>
<tr>
<td>Cache of mobile users</td>
<td>10 items</td>
</tr>
<tr>
<td>Cache of Access points</td>
<td>50 items</td>
</tr>
<tr>
<td>TTL value</td>
<td>500 s</td>
</tr>
<tr>
<td>Transmission range of Access Points</td>
<td>100 m</td>
</tr>
<tr>
<td>Transmission speed of Access Points</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>Transmission range of Mobile devices</td>
<td>10 m</td>
</tr>
<tr>
<td>Transmission speed of Mobile devices</td>
<td>2.5 Mbps</td>
</tr>
</tbody>
</table>

every message and spreads initially. If a node does not find the destination in the spray phase, it waits for the destination to perform direct transmission. In our experiment, Spray-and-Wait generated 10 copies for every message in the spray phase. First contact generates only one copy per message. The hybrid one sprayed Interest packets (limited to 10 copies) until the request reached the content providers and then used the Spray-and-Wait routing to deliver the content back to the requester.

B. QUERY DISTRIBUTION

We generate user interest based on the available contents C, which we assume are 1000 (i.e. |C| = 100). We assume that the i-th content c_i is the i-th most popular one π_i < π_j ∀ i ≤ j.

The users’ request profiles are randomly generated via the uniform distribution. Content popularity is correlated with user requests [34] and follows the well-known Zipf distribution [35]. In this work we consider two cases for the content popularity: (i) uniform and (ii) Zipf on initializing π, ∀c ∈ C. For the Zipf distribution we initialised the parameter to 1 and the normalizing constant to 0.2.

C. PERFORMANCE BOOST OF PROPOSED ARCHITECTURE

We measure the performance of our proposal using the metrics that are listed in Table 2. Figure 4 shows how the content requests are served of each type of DTN routing protocol. Practically, we show how the hit rates of each content provider type are related. Upon every request, the proposed mechanism uses all the possible ways in parallel to download the content as soon as possible. As we can see from both Figure 4a and Figure 4b the content caches in the CAPs
FIGURE 4. The service rate, i.e., the distribution of the content responses from each type of source for each DTN routing protocol. (a) Uniform content popularity. (b) Zipf content popularity.

FIGURE 5. Comparison of our proposal with other frameworks that do not support either CCN functionalities or caching. (a) Load in the content provider in the case of contents with popularity that follows the uniform distribution. (b) Load in the content provider in the case of contents with popularity that follows the Zipf distribution.

FIGURE 6. The benefit of CCN in the DTN routing protocols in terms of packet drop.

Together with the caches in the mobile nodes can handle more than 90% of the requests. Only in the case of the Spray and Wait routing protocol the requests are served by the content producer around 25% when the content popularity follows the uniform distribution and 20% when they follow the Zipf distribution.

In order to measure the contribution of the CCN mechanisms and the caches in the mobile users and in the CAPs, we implemented three simpler mechanisms and we compare them with our proposal in Figure 5. The first one is a simple content search using the DTN mechanism, i.e., that is operating as a request-response application on top of the DTN routing protocols and is denoted by DTN. The second one is an improved version of the first one that has content caches. Each content cache can store 10 objects. This mechanism is denoted by DTN with user caching. The last one is the same as our proposal but without caching in the mobile users and is denoted by Proposed Model without user Caching. As we can see from both 5a and 5b, our model is under-loading the content providers more that the other competitors. It is worth mentioning that in the case of Epidemic routing, the content provider is overloaded because unlimited number copies of each request is generated until the request reaches the content provider.

Next, we present in Figure 6 the benefit of using CCN mechanisms in conjunction with the DTN routing protocols because they filter the requests and stop forwarding identical packets. We observe that if user caching is not used, the number of duplicate packets significantly increase. This is happening because the request packets stay in the network longer to reach the potential content provider, and, hence the communication overhead in terms of additional traffic (interest/data packet) increases. Our proposal detects those duplicates and drops them accordingly. For instance, in the case of uniform distribution and multi-copy routing (e.g., Epidemic, Spray and Wait and Hybrid routing), we observe that more than 50% of the duplicate packets are reduced in our proposal as compared to our proposed model without user cache. This is happening because these protocols are producing multiple copies per request and each content has the same chance of being requested multiple times and being found in a nearby user’s pending interest table. On the other hand, in the case of First contact (single copy routing), we observe 27% duplicate packet reduction. In the case of Zipf content popularity and First Contact, we observe more than 60% duplicate packets reduction. This is because without the
user cache, the requests take a longer time to reach the potential content provider, whereas the other three DTN protocols can potentially reach the content provider faster than First Contact.

We also examine the service load on mobile users with our proposal to compare DTN with user caching. Figure 7 shows that our model reduces the service load on the mobile user in all routing. Especially in First Contact routing, the service load on the mobile user is significantly reduced by 57% when the content popularity follows uniform distribution. On the other hand, when Hybrid and Epidemic routing is used, the service load is reduced by 37% and 28% respectively. This is because First contact generates single copy for each request, whereas others use multiple copies. Multiple copies increase the probability of reaching the content provider faster. Service load is not significantly reduced (10%) by the Spray and Wait routing due to a limited number of message copies. We observe that in the case of Zipf content popularity, service load reduction (approximately 10%) on mobile users by Spray and Wait routing is almost similar to Epidemic routing.

Furthermore, we examine the changes in the average delay of the content retrieval in Figure 8. As expected, we had a decrease in the delay in most of the cases because of the caching mechanisms. Especially in the case of contents with popularity that follows the Zipf distribution, the contents were accessed faster because they were cached somewhere nearby. However, there are cases where the delay can be increased because there are not many requests for contents in the Spray and Wait routing protocol with contents that follow the uniform distribution (Figure 8c).

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigated the possibility of using mobile users in improving the performance of content delivery. For this, we explain the necessary required modifications in the conventional CCN mechanism in order for it to be functional in a DTN environment. Furthermore, we present a mathematical model of the content centric networking framework that exploits the opportunistic communications among mobile users. The proposed framework is implemented in ONE simulator to evaluate the concept. The simulation result shows that caching on mobile devices and cellular access points can improve the content retrieval time by more than 50%, while the proportion of the requests that are delivered from other mobile devices can be more than 75% in many cases. Our next steps will be focused on the development of caching policies and on various types of contents that are application dependent. Moreover, we plan to consider incentives that motivate mobile users to cooperate and store other content.

REFERENCES

HASAN M. A. ISLAM received the bachelor's degree in computer science and engineering from the Bangladesh University of Engineering and Technology, which is the top-ranked university in Bangladesh in 2008 and the M.Sc. degree in networking and services from the University of Helsinki in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Aalto University. His research interests include information centric networking, communication network architecture, and network protocols.

DIMITRIS CHATZOPoulos received the Diploma and M.Sc. degrees in computer engineering and communications from the Department of Electrical and Computer Engineering, University of Thessaly, Volos, Greece. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. He is also a member of the Symlab. His main research interests are in the areas of device-to-device ecosystems, mobile computing, mobile augmented reality, and cryptocurrencies.

DIMITRIJ LAGUTIN received the M.Sc. (Tech.) degree from the Helsinki University of Technology in 2005 and the D.Sc. (Tech.) degree from Aalto University, Finland, in 2010. He was a Researcher in several research projects with Helsinki University of Technology and Aalto University, including EU FP7 PSIRP and PURSUIT projects. He is currently a Project Manager and a Post-Doctoral Researcher with the EU Horizon 2020 POINT Project. His research interests include network security and privacy, future network technologies, and the Internet of Things. 


PAN HUI received the B.Eng. and M.Phil. degrees from the Department of Electrical and Electronic Engineering, The University of Hong Kong, and the Ph.D. degree from the Computer Laboratory, University of Cambridge. Before returning to Hong Kong, he spent several years in T-labs and Intel Research Cambridge. He is currently a Faculty Member with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, where he directs the HKUST-DT System and Media Laboratory. He also serves as a Distinguished Scientist of Telekom Innovation Laboratories (T-labs), Germany, and an Adjunct Professor of social computing and networking with Aalto University, Finland. He has published over 150 research papers and has some granted and pending European patents. He is an ACM Distinguished Scientist. He has founded and chaired several IEEE/ACM conferences/workshops, and has been serving on the organizing and technical program committee of numerous international conferences and workshops, including ACM SIGCOMM, the IEEE Infocom, ICNP, SECON, MASS, Globecom, WCNC, ITC, ICWSM, and WWW. He is an Associate Editor of the IEEE Transactions on Mobile Computing and the IEEE Transactions on Cloud Computing.

ANTI YLÄ-JÄÄSKI was with Nokia from 1994 to 2004, where he was involved in several research and research management positions, with focus on future Internet, mobile networks, applications, services, and service architectures. He has supervised 231 master’s theses and 21 doctoral dissertations during his professorship in Aalto University from 2002 to 2015. He is currently the Vice Head of the Department Computer Science, Aalto University. He has published over 100 peer-reviewed articles and he holds several international patents. His current research interests include mobile and cloud computing, crowdsourcing, and energy-efficient computing and communications.