Computer Communications 193 (2022) 214–223

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Context-based caching in mobile information-centric networks

Luís Leira [a,c], Miguel Luís [a,b,*], Susana Sargento [a,c]

[a] Instituto de Telecomunicações, 3810-193 Aveiro, Portugal
[b] ISEL - Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, 1959-007 Lisboa, Portugal
[c] Departamento de Eletrónica, Telecomunicações e Informática, Universidade de Aveiro, 3810-193 Aveiro, Portugal

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
Mobile networks
Information-Centric Networking
Named Data Networking
In-network caching
Context-based caching

ABSTRACT

Wireless networking is expected to sustain the direct interaction between personal users' devices, and to provide connectivity on large-scale resource-constrained devices. However, conventional networking protocols fail in large scale mobile wireless environments, due to node mobility, dynamic topologies, and intermittent connectivity. Information-Centric Networking (ICN) has been considered the most promising candidate to overcome the drawbacks of host-centric architectures where Named Data Networking (NDN) is one of the well-known and studied architectures within the ICN paradigm.

The main objective of this work is to improve both content availability and network performance in mobile environments regarding the ICN paradigm. This is provided through a context-based approach for the caching admission policy providing in-network caching and content replication, facilitating the efficient and timely delivery of information. Content popularity, freshness, proximity, source mobility type and network density are some of the metrics considered in the caching decision. We conducted a comparative study between our proposal and the NDN caching strategy by using two different datasets with real mobility and connectivity traces, addressing intermittent communication. According to our results, we observed that using a multi-criteria context-based cache admission policy improves cache hits, cache evictions, and request satisfaction ratios in mobile environments, thus improving content delivery and network efficiency.

1. Introduction

The rapid increase in the number of small embedded devices, along with the new wireless technologies, allowed simple day-life objects to compute and communicate, resulting in what is called the Internet of Things (IoT) [1]. Under the IoT concept, every device ranging from tiny, constrained sensors, to more powerful smartphones and networked vehicles, can be connected between each other and to the Internet. Devices interact with the physical surrounding environment to support a wide range of context-aware services, collecting and processing data to enable applications, machines, humans, and things to better understand their surrounding environments [2].

IoT addresses a wide range of applications; however, one of the most investigated vertical among the research community is the Smart City [3]. In this scenario, an extensive number of sensors and data generators (some of them placed in high mobile devices) are used to collect all types of information increasing the number of communicating elements. The consequence is an avalanche of mobile and wireless traffic information. The high heterogeneity and volatility of the network carries connectivity issues, such as long and variable delays, sparse and intermittent connectivity, high error rate, high latency, highly asymmetric data rate, or even a non-existent end-to-end connectivity [4].

The current Internet communication model is built according to a host-centric view of the network, following the traditional TCP/IP model. Such architecture was designed to connect a limited number of computers without considering native mobility support, in-network caching, and content-based security, to name a few. To fulfill all these needs, new approaches have been proposed [5]. Among them there is Information-Centric Networking (ICN), a different communication paradigm providing an alternative to the traditional end-to-end communication model of the current Internet by putting information dissemination at the center of the network-layer design [6,7]. Conceptually, in ICN, each piece of data has a unique, persistent, and location-independent name that is directly used by the applications for Content search and retrieval. Therefore, ICN enables the deployment of in-network caching and Content replication to improve the delivery of information. Additionally, the Content naming and Content-based security concepts adopted by the ICN architecture, allied to a location-independent data identification, provide efficient support for users' mobility. Furthermore, ICN provides data retrieval based on the request-reply exchange model, and native support of multicast.

Recently several ICN architectures have been proposed. One of the most promising and best-known architectures is the Named Data

* Corresponding author at: Instituto de Telecomunicações, 3810-193 Aveiro, Portugal.
E-mail address: nmal@av.it.pt (M. Luís).

https://doi.org/10.1016/j.comcom.2022.07.017
Received 3 February 2021; Received in revised form 8 January 2022; Accepted 6 July 2022
Available online 14 July 2022

Networking (NDN),[1] which originally proposes a pull-based communication model, hierarchical naming, in-network caching, and follows a Listen First Broadcast Later (LFBL) [8] approach. However, it was initially designed to cover infrastructure networks, without considering mobility and therefore new mechanisms for mobility support, especially focusing on caching routines, have been proposed [9,10].

In this work, we propose a cache admission policy to be incorporated in an ICN design, namely NDN, which is able to deal with mobile environments, and whose decisions take into consideration the context information of Contents, nodes, and the network. This cache admission policy, denoted as *Context-based Cache Admission Policy (ctxCAP)*, is based on parameters concerning the associated context: content popularity, freshness, proximity, and source mobility type. This caching proposal leads to several changes and additions to the current NDN architecture, such as the maintenance of node properties like the node mobility type and its coordinates. This approach will lead to a more rigorous and careful decision process about the Contents to be stored, thus enhancing the performance of caches, and consequently, of the entire network operation.

The proposed *ctxCAP* has been evaluated in a mobile network, using two real and different mobility and connectivity datasets, addressing intermittent communications use cases. Several scenarios have been considered, changing the role of each network element concerning the NDN concept: they can be consumers, producers, or both. Different types of Contents have been considered, each one with a specific Content freshness and request rate. Based on the obtained results, it is observed that *ctxCAP* provides significant benefits in terms of cache hits and evictions ratio, and request satisfaction ratio in these mobile environments, with consequent strong benefits for both NDN performance and users' satisfaction.

The remaining of the paper is the following. Section 2 motivates this work by introducing the background concepts related with NDN, detailing the caching-related schemes and discussing the related work. Section 3 presents the proposed context-based caching mechanism and the modifications introduced to achieve this approach. In Section 4, we present and discuss the experimental results. Finally, Section 5 enumerates the conclusions and overviews the future work.

2. Motivation

2.1. Background

The work here proposed follows the well-known NDN architecture, where two types of packets ensure the ICN communication: the **Interest** packet, used to define a request for Content, and the **Data** packet, containing the data itself. Both have a common Content identifier designated as *Name*, which is a sequence of name components that follow a hierarchical structure with variable length.

The devices exchanging information in the NDN architecture are treated as nodes that integrate the NDN logic. They can behave either as producers (entities originating and signing Contents), consumers (entities that issue interests for certain Contents) or router/intermediate nodes (entities which forward interest packets and route data packets between consumers and producers). A node can assume these three roles.

An NDN node maintains three basic structures known as Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). The first one is a cache for incoming Data packets, which may be used for faster replies to Content requests. The second one is a table of Interests and respective 'faces' from which they came and that are waiting for the requested Data. The last one is a table of name prefixes with respective outgoing 'faces', which allows the routing of the Interest packets through a name-based lookup. The component known as 'faces' enables to receive and forward packets; thus, it is seen as an interface with appropriate mechanisms.

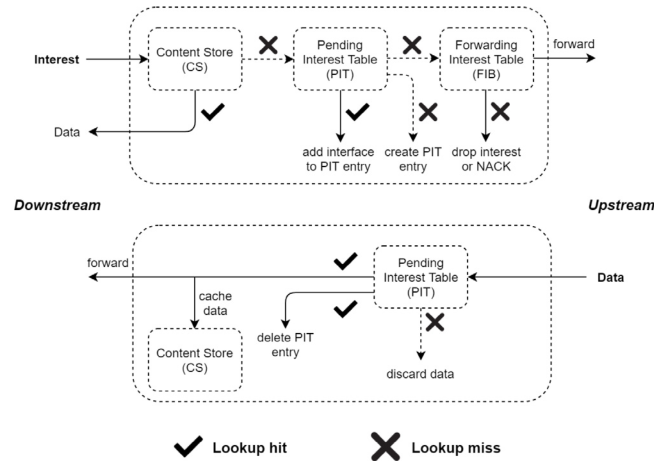

Fig. 1. Flow of Content through the network in NDN.

Each Interest packet contains the name of the Content to achieve alongside with any useful network data relevant information to the future intervening nodes. It is sent to the network, being forwarded along established paths that are present in its FIB. When an Interest reaches a node, it checks its CS for the Data packet. If it is not in the CS, it follows to the PIT where it stores the Interest and the interface from which it arrived, and then makes use of the FIB by routing the Interest packet, if possible. When the Interest reaches a node that has the requested data, that node replies with a Data packet containing the corresponding Content, along with a signature done by the producer. This Data packet follows the path taken by the Interest packet in reverse, returning to the consumer, which made the request. This also applies to the Interest NACK packets that notify errors like congestion, interest duplication, or no route found for that data packet, which also follows the taken path to the consumer. Fig. 1 illustrates the flow of the Interest and Data packets, such as triggering mechanisms on each node.

NDN uses a link protocol known as NDNLP version 2,[1] It operates as a link adaptation layer, located between link and network layers. The NDNLP packet, known as *LpPacket* carries NDN packets and can also contain additional information, which is represented by tags encoded into the header field. Being a single-hop protocol and with the forwarding plane not obliged to preserve tags, it is beneficial to use tags in cases where it is necessary to update these parameters along a path. The Interest and Data packets are then encapsulated using the link protocol NDNLP, as represented in Fig. 2.

2.2. Caching in ICN

Caching is about storing Content temporarily to serve it on future requests, reducing the cost of transmitted messages by either both producers and consumers. Such traffic reduction may also help in terms of energy consumption. It can also reduce a response bottleneck by having a Content distributed by the network, increasing availability, and decreasing latency [11,12]. However, this can be complex as it is necessary to decide where to place Contents and how to manage them when the cache needs to evict some Content over another, or how to maintain cached Contents consistency. It is also essential to determine if it is more advantageous to perform storage in caches along the path (*on-path caching*) or any other caches (*off-path caching*).

A caching strategy can have different characteristics according to three different phases: placement, replacement, and coherency.

[1] http://named-data.net/.

[1] https://redmine.named-data.net/projects/nfd/wiki/NDNLPv2.

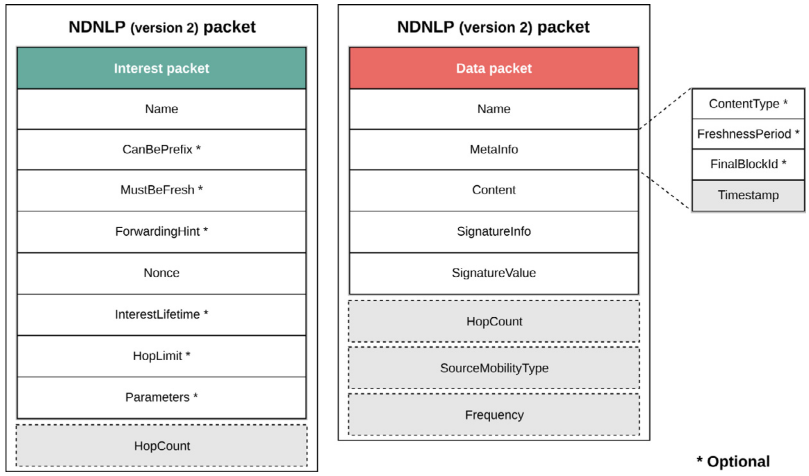

Fig. 2. Communication packets format (original NDN fields marked in white).

Placement is about deciding where Content should be stored. This choice is based on a Content placement scheme like Leave Copy Everywhere (LCE) [13], Edge-Caching (EC) [14], Client-Cache (CC) [15], among others, such as Leave Copy Down (LCD) [16], Between-Centrality Caching (Betw) [17], Ego Network Betweenness Centrality (EgoBetw) [17], and Probabilistic Caching (ProbCache) [18]. The LCE strategy indicates that all nodes through which Content goes through will store it. EC indicates to store only on the previous-hop node regarding the consumer. CC proposes to store Content in nodes with a client attached.

Caching placement schemes for ICN can be sub-categorized into three types:

- **Content Based Caching (CBC)** uses Content properties to decide what Content should be stored.
- **Context and Node Based Caching (CNBC)** uses Content and node internal properties.
- However, there are still other proposals that fit the **Alternative Caching** sub-category, considering infrastructure-based caching and cache Contents coherency.

Replacement is about determining which cached Content should be replaced by a new Content. There are several Content replacement schemes like First-In-First-Out (FIFO), Least Recently Used (LRU), among others, such as Least Frequently Used (LFU) and Random (RR). In FIFO, the Content that came in first of all is chosen as the cache victim to be replaced. LRU selects the Content that has not been used more recently, which exploits temporal locality.

Each device has memory limits, and therefore, limited size caches. For this reason, when a cache is full, it is important to decide which Content will be replaced by new Content. Caching replacement schemes for ICN can follow a classification that matches largely existing operating system paging and web cache policies [19], so that they can be sub-categorized into five types: Recency-based policies which keep the recently referred Contents in the cache (e.g. LRU); Frequency-based policies which keep the most requested Contents, that is, the most popular Contents (e.g. LFU); Randomized policies which select the cache victim randomly, thus avoiding the overhead caused by calculations (e.g. RR); Size-based policies which considers that larger Contents are eliminated first (e.g. SIZE); Alternative policies which may incorporate and combine features of the previous mentioned, or take into account parameters in an ICN context (e.g. LFRU [20], which uses an LFU approximation and a partitioned LFRU, being divided into two partitions, and Universal Caching [21], which considers the distance between the Content provider and the consumer, the request frequency of the Content and the number of outgoing links).

Coherency is about checking the validity of the cached Contents. Coherence mechanisms are focused on verifying if the Content is still up to date.

2.3. Related works

In the last years efforts have been devoted to the development of novel caching strategies capable of increasing the request satisfaction ratio, while reducing the number of hops traveled as well as the number of cache evictions. Efficient caching strategies can be quite different as they depend on the requirements, which rely on the network characteristics, fixed versus mobile, and on the type of application, e.g. video content delivery, IoT information acquired from sensors or even real-time communication. To name a few works about caching strategies, the work in [22] proposed a content placement strategy in a static network, entitled Greedy Caching, that adopts a locally optimal approach at each node to determine the set of Contents to be cached at each node based on relative content popularity, which is calculated based on the request miss stream from downstream nodes. Simulation results show the effectiveness of the proposal but, as mentioned before, no mobility is considered.

The work in [23] discusses several existing ICN caching and cache replacement strategies in an IoT context where, for example, the IoT nodes' memory is often limited. The results, obtained through experimentation using real IoT hardware, have shown that stateless caching policies can perform equally well, or sometimes, even better than more complex and statefull schemes. In [24], authors suggest a probabilistic caching strategy (known as pCASTING) that considers the Contents freshness and also node properties like battery level and cache occupancy. In [25], it is proposed a scheme using Content freshness, the request rate (meaning the popularity of each Content), and the node properties like the incoming request rate and the node location in the network. Still in the IoT context, the work in [26], uses specific freshness values together with a decision strategy with dynamic probability (always and probabilistic with $P = 0.5$) for the caching policy. In [15], it is presented a strategy for Machine-to-machine (M2M) based on client caching together with a cache coherence algorithm. It has a preference for the nearest consumer nodes to perform caching, while using the coherence algorithm to check the validity of the Contents. The work in [27] presents Near Cache Placement (NCP) for IoT-based traffic class, which deals with a multi-objective optimization problem, whose heuristic moves requested Contents closer to consumers by decreasing the delivery cost between the original provider and the replica-node, and between the replica-node and the consumers, besides the caching cost in replica-nodes. In [28], it is proposed Approximate Betweenness Centrality (ABC), which uses topology-based heuristics from existing strategies ($Betw$ and $EgoBetw$), and does not need previous knowledge of the network. In [29], it is proposed a Periodic Caching Strategy (PCS) that deals with a statistics table which records the most frequently requested content by using a threshold, the frequency count of each content, and the last requested time, to rank contents in terms of popularity and recency.

On the mobile plane, caching strategies have been focused mostly on cellular technologies. The work in [30] analyzed the feasibility of having ICN integrated into LTE mobile networks, and the results have shown that its inherent caching features enable lower latencies to access content and reduce traffic at the core network at the cost of a low impact on the evolved Node B performance. The works [31,32] explore different caching strategies for mobile access networks. While the first one follows a strategy driven by the point of view of a network operator seeking to minimize investment by realizing an optimal tradeoff of cache memory for network bandwidth, the latter proposes a proactive caching with redirection approach to enhance the support of seamless mobility within ICN networks as well as the Quality of Service. Lately, the adoption of ICN on 5G networks has received increasing attention. In [33] an ICN-based caching approach for mobile video in 5G networks has been proposed where user mobility is exploited to reduce the retrieval delay caused by frequent handovers. Also targeting a 5G architecture, the work in [34] proposed a QoE-driven DASH video caching and adaptation at mobile edge where the focus is to provide assurance to user Quality-of-Experience against dynamic radio access network conditions, which is achieved through context-aware caching and serving the DASH segments at their appropriate quality levels.

The works mentioned before do not admit the direct communication of mobile nodes, *i.e.* the communication with the Base Station must always occur. In device-to-device capable networks the caching procedure in mobile networks is even more important since Contents cached in mobile elements can be achieved directly from other mobile nodes, therefore increasing the network performance. The work in [35] discusses and evaluates the limitations and opportunities of using ICN in Vehicular Ad-hoc Networks (VANETs), comparing different strategies on in-network caching. The results have shown that mobility takes an important role in the network performance, and eager caching strategies do not necessarily lead to improvements in content delivery. In [36] a proactive caching approach for ICN-based VANETs is proposed where parameters such as the geolocation, the current velocity and the heading direction of the mobile elements are considered for the caching decision. Although considering important metrics, as the ones mentioned before, the proposal does not take into consideration parameters related with the Contents, such as freshness and/or the Interest frequency. A similar approach is used in [37] where the expected mobility of autonomous vehicles is used in the caching decision of video-related Contents.

The integration of cache placement with replacement policies may increase or degrade the efficiency of a caching strategy, so it is necessary to reconcile the best decisions of both placement and replacement policies, as well as ensuring coherency of the cached Content. None of the previously presented works on placement strategies take into account high dynamic mobile scenarios with continuous topology changes. These decisions must consider the mobility of the nodes, in addition to the combination of several relevant attributes taken from the context associated with Contents and nodes. As many of the mobile devices are considered somehow constrained, it is important to have replacement policies that can perform efficiently, having a preference for low complexity, and also complement the placement strategy.

3. ICN with context-based caching

The NDN original caching design follows an LCE placement strategy: all nodes that are in the path between the consumer and the node serving the requested Content insert the Data packet in their cache, expecting to satisfy future requests. This strategy caches Content without any criterion, which can cause several problems such as Content redundancy, negative influence on nodes' energy efficiency, besides increasing the number of evictions.

In this paper we propose a cache admission policy whose function is to evaluate if the Content has enough potential for satisfying future requests, justifying its insertion in the CS. It will influence the placement of Content and will evaluate whether it is beneficial to replace other Contents in the cache when it is full. The cache admission policy will replace the placement strategy and assist in the replacement strategy, and is denoted as *Context-based Cache Admission Policy (ctxCAP)*.

This strategy requires some modifications to the base work which are presented below:

- **Interest packet:** one additional field in NDNLP packet to record the number of hops (*hop-count*), shown in Fig. 2;
- **Data packet:** one additional field to record data creation timestamp (*timestamp*), and three new fields in NDNLP packet to record the number of hops (*hop-count*), the number of times Content was requested (*frequency*), and the mobility type of the node serving the Content (*source mobility type*), as shown in Fig. 2;
- **Node properties:** two additional fields to record the node mobility type (*node mobility type*) – so it can be used to define the *source mobility type* data packet parameter – and mobility prediction (*mobility prediction*);
- **Content Store:** two additional fields to record the number of times the Content was requested locally (*local frequency*), and the average proximity from consumers who requested the Content (*proximity average*).

The Content Store has a leading role in two different situations:

1. The Interest packet is received, and the node has to verify if the Content is cached;
2. The Data packet is received, and the node must decide whether to cache the Content.

In the first case, in the process of verifying if the Content X is in the CS, if this is true, the CS entry will be updated so the *frequency* (fq_X) and *local frequency* ($localFq_X$) are incremented by 1 and *proximity average* ($pxAvg_X$) is updated as

$$pxAvg_X = pxAvg_X + [((1/ihc_X) - pxAvg_X)/localFq_X], \quad (1)$$

where ihc_X is the hop count of the interest packet for Content X. The initial value of $pxAvg_X$ is 0, while the initial value of $localFq_X$ is 1. These are relevant parameters since they are used in the cache admission policy.

In the second case, when a node receives a new Content Y, it will have to decide if it caches the Content or not. It will submit the Content to the *ctxCAP*, which will determine if the Content should be admitted to the CS. If the cache is not full, the admission occurs if the returning value P, obtained by the ctxCAP algorithm, and explained further ahead, is equal or greater than a certain threshold T, as represented in Fig. 3. Otherwise, the CS will use the replacement strategy to select a cache victim (Content Z), applying the *ctxCAP* to that Content and finally comparing it with the new one. The Content with the highest P value resulting from the algorithm will be cached, and the other will be discarded. This last phase, known as replacement, is represented in Fig. 4.

The *ctxCAP* is based on Content, node and network properties, which are represented by the following parameters:

- **Popularity:** it is important to consider the Content popularity, since a more popular Content is more likely to be requested. The popularity of a Content X, represented by PO_X, is obtained by dividing the number of times it has been requested, fq_X, by the number of requests satisfied by the most solicited Content in the cache, $maxFq$, *i.e.*,

  $$PO_X = \left(fq_X + 1\right) / (maxFq + 1), \quad fq_X \geq 0 \wedge maxFq \geq fq_X, \quad (2)$$

  with $PO_X \in [0, 1]$.

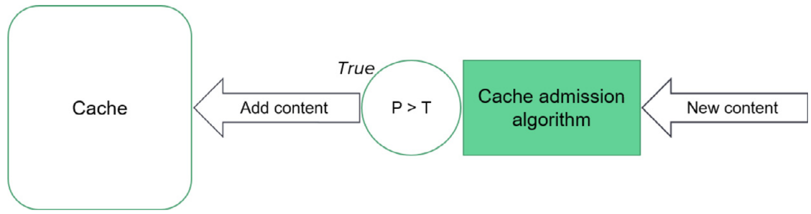

Fig. 3. Placement diagram considering that cache is not full, and admitting the Content if the result is greater than a threshold T.

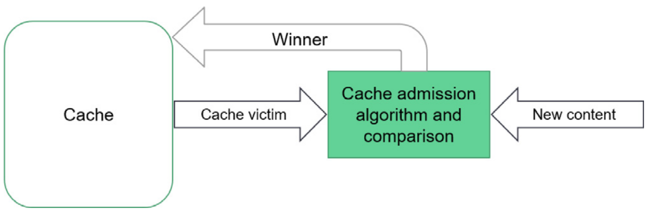

Fig. 4. Replacement diagram considering that cache is full and Content is admissible, whose replacement policy will return a cache victim and will go through the cache admission policy to decide which Content will be in cache, being called the winner.

Algorithm 1: Cache admission probability algorithm for a given Content.

```
Input: newContent, freq, maxFreq, cTime, pTime, f, pxAvg, sm, hopCount, ihopCount
Output: P
Procedure admission probability
PO ← (freq + 1)/(maxFreq + 1)
FR ← 1                                  // 1 by default (f_X = 0)
if f_X > 0 then
    FR ← 1 − [(cTime − pTime)/f]
    if FR ≤ 0 then
        return 0                        // Content is obsolete
if !newContent then
    PX ← pxAvg              // Average proximity from previous hits
else
    if sm = 1 ∧ hopCount = 1 ∧ iHopCount > 2 then
        PX ← 1 − (1/iHopCount)
    else
        PX ← 1/iHopCount
return P = (PO ∗ FR ∗ PX)^(1/3)        // Returns the cache admission
  probability
```

- **Freshness:** fresher Contents may be beneficial because they present a longer lifetime percentage that guarantees their validity, which means that they are more likely to be requested. The freshness of a Content X, represented by FR_X, depends on the non-negative freshness value associated, given by f_X from the FreshnessPeriod field in the NDN Data Packet. If f_X equals zero, it does not have a validity value associated, so it is always fresh, resulting in $FR_X = 1$. For the remaining cases, FR_X is calculated as

$$FR_X = \begin{cases} 1 - \left[(cT - pT)/f_X\right], & f_X > 0 \\ 1, & f_X = 0 \end{cases} \tag{3}$$

  for $cT \geq pT$. cT represents the current time and pT the Content's produced time. $FR_X \in [0,1]$ if it is a valid Content; otherwise, it will indicate that the Content is obsolete, being soon discarded.
- **Proximity and Source mobility type:** the distance between consumers and producers may be a relevant parameter to take a better placement decision. Besides the Content hop-count, represented by hc_X, there is also a variable that represents the Interest hop-count for a given Content, given by ihc_X.
  In addition, the fact that a mobile or a fixed node provides a specific Content may also be relevant because mobile nodes can cause sudden topology changes, so the placement decision process should, therefore, take this into account. The source mobility type (sm_X) can take two values: 0, when the Content provider is static, or 1 otherwise.
  Thus, the proximity of a Content X, represented by PX_X, in the case of a newly arrived Content can be obtained as follows

$$PX_X = \begin{cases} 1 - (1/ihc_X) & \text{if } sm_X = 1 \wedge hc_X = 1 \wedge ihc_X > 2 \\ 1/ihc_X & \text{otherwise} \end{cases}, \tag{4}$$

  where $PX_X \in [0,1]$. If the Content already exists in the CS, then PX is given by $pxAvg_X$, detailed in (1).

A complete caching scheme contemplates placement, replacement, and coherency mechanisms. These are conditioned by the *ctxCAP* whose parameters described above influence and impact on the three phases. Algorithm 1 describes the computation of the caching admission probability, whose result, P, will be used to decide whether to cache. It is assumed that each parameter (popularity, freshness and proximity and mobility type) has the same weight on the final decision.

The caching decision, either placement or replacement, is based on the result of the *ctxCAP* algorithm. The output of the algorithm can be simply used in the replacement process to decide if the newly arrived Content should replace an existing one in the CS if the P value of the new Content is bigger than any other in the CS. On the other hand, if the CS is not full, the approach is to cache only in places of high demand. The objective is to place the Contents with higher *significance*, higher P, in highly populated regions. For that, this work assumes the existence of a mobility management system responsible for analyzing the node's mobility, returning mobility-related characteristics such as the density of the network zones and estimations about the node's next positions [38,39].

The node's mobility can be characterized and predicted, and depending on the accuracy of the predictions, it can be used to understand if a given node will be in a specific area, more or less popular, *e.g.* characterized by the number of wireless connections. Being in a popular region will translate into a probability value, which will be directly used as a threshold T ($T \in [0,1]$) for Content admission. The idea is to divide the scenario in cells/regions and, for a given period of time, a network density classification is performed [39]. Higher values of T means that a mobile node will likely be in one of the most highly populated regions (fewer regions). Lower T values are less restricted in the region characterization, including both less and more populated regions. By other words, the threshold will differentiate zones with higher network densities from those where, historically, there are a small number of network elements. By placing the Content in mobile nodes in high density regions, we are increasing the probability of availability and reachability of that Content.

4. Experimental results

In this section, we assess the performance of *ctxCAP* in a large scale vehicular network. We also perform a comparative evaluation with the NDN base version, which implements *LCE*, resorting to $ndnSIM$.[2]

[2] NS-3 based Named Data Networking (NDN) simulator (https://ndnsim.net/current/).

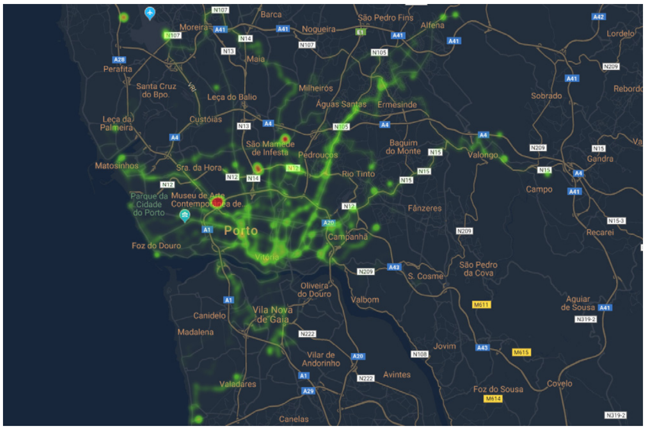

Fig. 5. Heatmap of mobility pattern of 40 buses (from 9 am to 1 pm on 23/01/2018) and 40 taxis (from 9 am to 1 pm on 23/01/2014), adapted to the same time interval.

4.1. Metrics

The performance of the context-based caching proposal and its heuristics will be evaluated by the following metrics:

- **Cache hit ratio** measures the average ratio of the number of cache hits to the number of cache lookups (number of cache hits plus the number of cache misses);
- **Cache eviction ratio** measures the average ratio of the number of cache evictions to the number of cache inserts;
- **Hops** measures the average number of hops that a data packet traveled from the Content provider to the consumer in order to satisfy a request;
- **Request satisfaction ratio** measures the average ratio of satisfied requests, *i.e.* when the Content reaches the consumer;
- **Response latency** measures the average delay between the moment that a node sends a Content request and the moment it receives that Content.

4.2. Scenario description

The evaluation scenario is composed of two datasets: the first one contains mobility traces of STCP buses,[3] and the second one contains taxi mobility traces, both in the city of Porto. Regarding the buses dataset, only 40 nodes were considered, using a time window of 4 h (from 9 am to 1 pm on 23/01/2018). The second dataset was taken from the Taxi Service Trajectory Prediction Challenge 2015[4] and, once again, only 40 nodes were considered for the same time window, from 9 am to 1 pm on 23/01/2014. Since the dates do not match it was required to adapt the datasets to match the same time frame (Fig. 5). The scenario assumes a set of Road-Side Units (RSUs) and backend routers spread throughout the city of Porto to connect the mobile nodes to Internet. Considering the RSUs' locations, there are certain areas with no coverage by the RSUs, which may make it challenging to communicate with the mobile nodes.

The mobility datasets used were also analyzed in order to make a profile about the vehicular density. The city map was divided into cells, each one covering approximately an area of 3.06 km^2 (1750 × 1750 m). Thus, following a time window of 15 min, vehicular density values were obtained for each cell. The result was a characterization of the number of vehicles per cell, represented by a value between 0 and 100, with 0 representing cells without vehicles and 100 representing a cell containing all the vehicles in the network. The use of these values aimed to analyze how nodes mobility and density can influence the *ctxCAP* threshold, and consequently, Contents admission decisions.

[3] https://www.stcp.pt/.

[4] http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html.

4.3. Simulation setup

Each node can have one or more communication interfaces. There are two types of On-Board Units (OBUs), depending if they are equipped in buses or taxis. A bus OBU has an IEEE 802.11p interface while a taxi OBU has an IEEE 802.11n interface. Each RSU has both IEEE 802.11p and IEEE 802.11n interfaces beyond a point-to-point interface with a backend router. All the IEEE 802.11p interfaces have a range of 400 m, and IEEE 802.11n interfaces have a range of 200 m, considering an urban environment [40]. They use the same transmission rate for every packet sent, following a constant rate transmission, with a channel data rate of 6 Mbps for IEEE 802.11p and 54 Mbps for IEEE 802.11n, whereas the latter also acts in *ad-hoc* mode. The point-to-point communication channels have a data rate of 1 Gbps and a delay of 1 ms.

Table 1 details the system parameters regarding the experiments. Each Data packet represents one Content chunk with a size of 1000 bytes, with no differentiation according to the nature of the Content. The Content requests popularity follows a Zipf–Mandelbrot distribution, one of the most used distributions in this type of performance analysis. The OBUs, RSUs, and backend routers caches have a size of 20, 60, and 100 entries, respectively. The cache decision policies (placement) considered for comparison purposes are *LCE* and *ctxCAP*, where the latter varies its threshold T between 0.25, 0.5 and 0.75 unless otherwise instructed. Because real traces of vehicular mobility and connectivity were used, we executed each experiment a single time. Still, in order to successfully compare the different results among the several scenarios and variations, we kept the same experimental conditions, *i.e.* the periodicity of the Content request and the Content to be requested.

The replacement policies considered to be used together with the cache placement policies are Priority FIFO (PFIFO) and LRU (e.g. *ctxCAP+PFIFO* or *LCE+LRU*). Because no routing protocol is used in this work, the Forwarding Strategy used is the *Best-route modified*, which follows the logic of forwarding the Interest packet through the least-cost known interface, and if it does not know the path to the Content, it sends it in multicast to all the nodes within reach.

The Contents on the network were divided to represent five common types: videos, pictures, music, environmental information, and waste containers information. The freshness of each Content is set by its type, such as the request rate from each consumer, which is defined in Table 2. All the consumers generate Interest for all produced Content with an *InterestLifeTime* of 2 s.

Two scenarios were considered for the evaluation process. In Scenario 1, 40 OBUs are consumers while the other 40 OBUs and 26 RSUs are producers. The 4 backend routers are only intermediate nodes. Regarding the consumers, half of them are buses, and the other half are taxis, so the same applies to producers. In this case, two Variants are considered: (1) 33 groups of producers, where there is a pair of RSUs or OBUs with the same Contents, which means that there are 2 replicas of each Content (1050 different Contents); (2) each one of the 66 producers generates unique Contents, which means 1 replica of each Content (2100 different Contents).

Scenario 2 follows the same characteristics as in Scenario 1 Variant 2, but in this case, it uses a fine tune version of the use of the cache admission policy threshold. The OBUs, RSUs, and backend routers locations define how its threshold will change according to the corresponding cell density value, obtained as in [39]. However, in this scenario, the network density value will only affect part of the decision threshold T: a default threshold of 0.7 is assumed to all regions, and the density value will affect the remaining 0.3 in two different strategies: in a first strategy, nodes in high-density cells will increase the threshold T, while in the second strategy, nodes in high-density cells will lower the threshold T. Regarding the use of 0.7 to be the default threshold value, as mentioned in the manuscript, it was the value that, after preliminary studies, resulted in the best performance of our heuristic. The default threshold value was found to be the best after preliminary studies.

Table 1
System parameters used in the simulation.

Parameter	Value
Total number of nodes	110
OBUs, RSUs and backend routers	80, 26 and 4
Content size	1000 bytes
Content popularity	Zipf–Mandelbrot distribution (s = 0.7, q = 0.7)
Cache size (OBUs, RSU, backend routers)	20, 60 and 100 entries respectively
Cache decision policy (placement)	LCE, ctxCAP(0.25), ctxCAP(0.5), ctxCAP(0.75)
Cache eviction policy (replacement)	PFIFO, LRU
Forwarding strategy	Best-route modified
Propagation delay model	ConstantSpeedPropagationDelayModel
Propagation loss model	RangePropagationLossModel
Simulation time	14 400 s

Table 2
Producers content freshness and consumer request rate by Content type for the traffic in Porto scenario.

Content type	Content freshness (s)	Request rate (Interests per second)
Videos	432 000	1
Pictures	86 400	1/2
Music	259 200	1/3
Environmental	600	1/4
Waste	3600	1/5

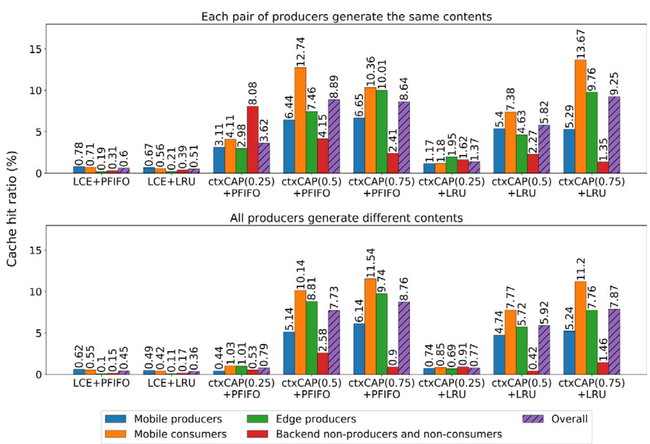

Fig. 6. Cache hit ratio versus different types of nodes and overall by cache combination for Scenario 1.

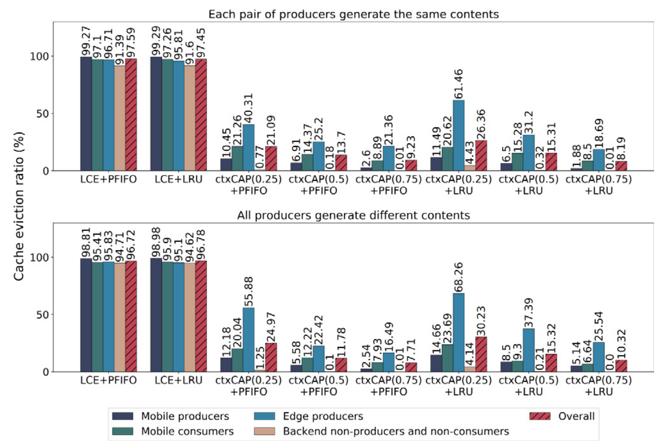

Fig. 7. Cache eviction ratio versus different types of nodes and overall by cache combination for Scenario 1.

Table 3
Request satisfaction ratio for Scenario 1.

Cache combination	Request satisfaction ratio (%)			
	Intermediate and consumer nodes		Only consumer nodes	
	Variant 1	*Variant 2*	*Variant 1*	*Variant 2*
LCE+PFIFO	39.59	20.18	1.55	0.89
LCE+LRU	28.64	1.03	1.49	0.86
ctxCAP(0.25)+PFIFO	68.82	66.85	2.01	1.21
ctxCAP(0.5)+PFIFO	72.0	71.23	2.32	2.62
ctxCAP(0.75)+PFIFO	**74.68**	**73.45**	**2.78**	**2.88**
ctxCAP(0.25)+LRU	68.45	62.62	1.71	1.1
ctxCAP(0.5)+LRU	70.67	67.02	1.88	1.27
ctxCAP(0.75)+LRU	73.72	70.75	2.59	1.55

4.4. Results

The following are the experimental results of the various caching combinations, where each one represents the use of placement and replacement strategies together, as explained previously.

4.4.1. Scenario 1

The *ctxCAP* strategies achieve a higher overall cache hit ratio compared to *LCE* strategies, mostly by a significant difference (Fig. 6). It is also noticeable that, typically, strategies that achieve higher cache hit ratio have lower cache eviction ratio, which is also justified by the threshold used: the higher it is, the fewer replacements will occur. A bigger cache eviction ratio is observed on edge producers and mobile consumers (Fig. 7). RSUs' eviction ratio can be justified by the fact that they are strategically positioned and are a common point of contact for many mobile nodes, and they have a larger cache; OBUs (mobile consumers) retain Contents requested by them or neighbors (acting as an intermediate node in this case), and both may request one of these cached Contents.

ctxCAP strategies serve Contents at a higher number of hops than *LCE* strategies, so the higher the threshold applied, the higher the percentage of Content that has traveled more hops (Fig. 8). Since consumers are always the same and located at the last level of the network, with *LCE*, there is a higher probability for the Content to be closer, while *ctxCAP* may be caching the Contents in supposedly more suitable nodes to increase the satisfaction of more requests (Table 3). The *ctxCAP(0.75)+PFIFO* has the best results, concerning the request satisfaction ratio with intermediate and consumer nodes; the difference to the baseline increases with the increase in the number of different Contents.

Because satisfied Content traverses more hops in *ctxCAP* strategies, the response latency will also increase, as confirmed by the results (Table 4). The differences are quite visible, as *LCE+LRU* clearly achieves the best results. The reason for *ctxCAP* additional delay is that it can retrieve the Content that was never found before. This means that it is increasing the cache hits at the cost of additional delay. It is also perceptible that *PFIFO* obtains better results when compared to *LRU*, which is justified by the more careful selection of the Contents to evict, giving preference to stale Contents, which does not happen with *LRU*.

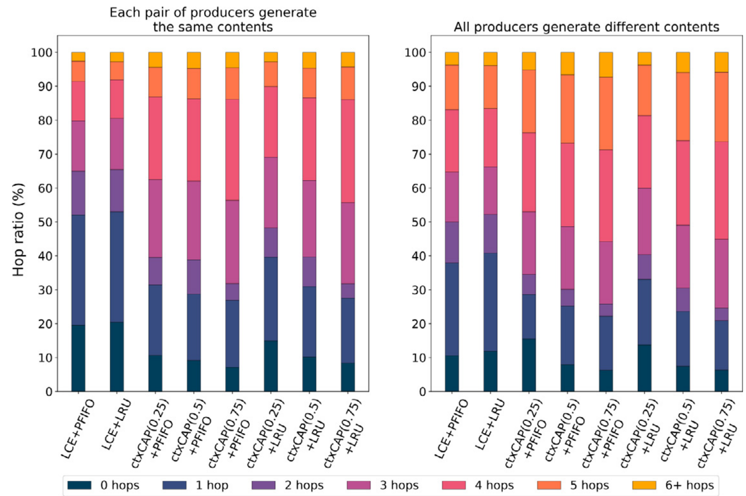

Fig. 8. Hops versus different types of nodes and overall by cache combination for Scenario 1.

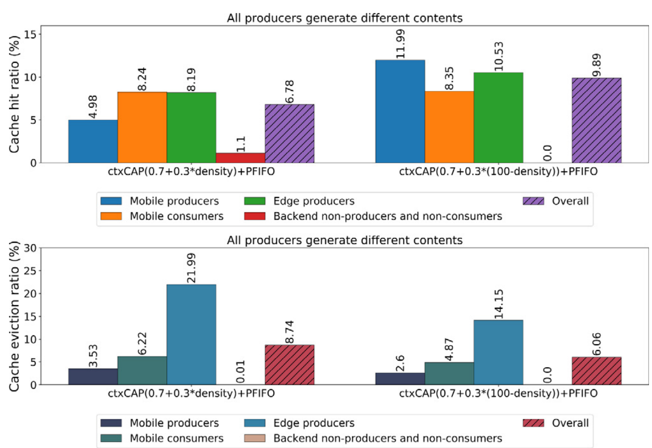

Fig. 9. Cache hit and eviction ratios versus different types of nodes and overall by cache combination used for Scenario 2.

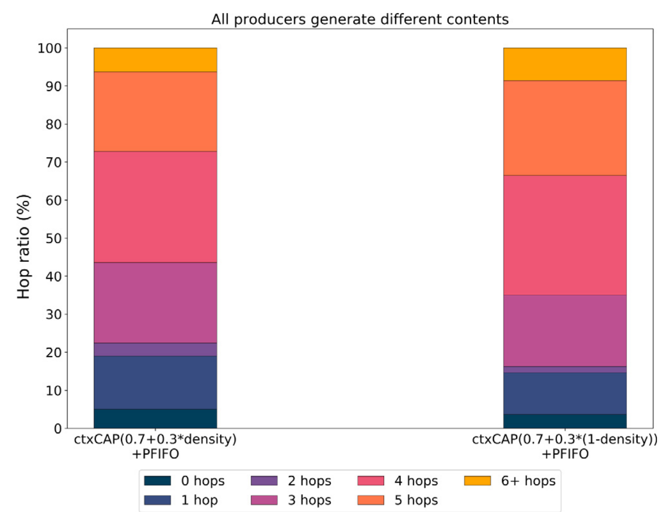

Fig. 10. Hops versus different types of nodes and overall by cache combination used for Scenario 2.

Table 4
Response latency for Scenario 1.

Cache combination	Response latency (s)	
	Variant 1	*Variant 2*
LCE+PFIFO	0.24	0.14
LCE+LRU	**0.14**	**0.04**
ctxCAP(0.25)+PFIFO	0.62	0.63
ctxCAP(0.5)+PFIFO	0.69	0.7
ctxCAP(0.75)+PFIFO	0.74	0.7
ctxCAP(0.25)+LRU	0.58	0.56
ctxCAP(0.5)+LRU	0.63	0.66
ctxCAP(0.75)+LRU	0.69	0.72

Table 5
Request satisfaction ratio for Scenario 2.

Cache combination	Request satisfaction ratio (%)	
	Intermediate and consumer nodes	Only consumer nodes
	Scenario 3	*Scenario 3*
*ctxCAP(0.7+0.3*density)+PFIFO*	71.04	1.78
ctxCAP(0.7+0.3(100-density))+PFIFO*	**76.27**	**3.43**

4.4.2. Scenario 2

Fig. 9 presents the cache hit and eviction ratios of the *ctxCAP* heuristic when we assume a different approach for the decision threshold T. The first observation goes to the increase in the cache hit ratio when compared to the best approach with the *PFIFO* strategy, achieved with the fixed threshold of 0.75. The overall increase is justified by the increase in the number of cache hits of mobile producers, as well as edge producers, although the difference in the last case is less significant. It is also observed that the higher cache hit ratio leads to a decrease in the cache eviction ratio.

Also, the higher the threshold, the higher the number of hops traversed by Contents, already expected as observed in prior results (Fig. 10).

ctxCAP(0.7+0.3(100-density))+PFIFO* gets the best request satisfaction ratio results in its two analysis (considering intermediate nodes and consumers, or only consumers). These results are even higher than the obtained with *ctxCAP(0.75)+PFIFO*, improving on 3.84% with intermediate and consumer nodes, and 19.10% with only consumer nodes (Table 5).

*ctxCAP(0.7+0.3*density)+PFIFO* response latency is slightly better, however, worse than that with the *ctxCAP(0.75)+PFIFO*, as previously noted (Table 6).

Regarding the *ctxCAP(0.7+0.3*(100-density))+PFIFO*, we noted that, for the 40 mobile producers, the average threshold of 8 of these nodes was 0.7, and 1.0 for the remaining 32 mobile producers. For the 40 mobile consumers, 2 of these nodes had an average threshold of 0.7,

Table 6
Response latency for Scenario 2.

Cache combination	Response latency (s)
	Scenario 3
ctxCAP(*0.7+0.3*density*)+*PFIFO*	**0.77**
ctxCAP(*0.7+0.3**(*100-density*))+*PFIFO*	0.78

while in another 2 nodes, it was 0.95, and the remaining was 1.0. For the 26 edge producers, 3 of these nodes had an average threshold of 0.7, and the remaining had 1.0. For the 4 backend routers, the average threshold was 1.0.

The nodes whose average threshold was 0.7 could be the ones traveling in low-density zones, so their threshold has remained unchanged. The fact that the overwhelming majority has an average threshold value of 1.0 is due to the low influence of nodes density values, since its value represents only 0.3 of the threshold value. However, these results show that the producer nodes, especially the mobile producers, have been harmed because they create a large number of Contents compared to their cache size. Since each producer creates different Contents, if they delete one Content to the detriment of another, it may be deleting the only replica of that Content, which results in a lower cache hit ratio, and hence lower request satisfaction ratio.

It could be beneficial to split the cache into two parts (one for Contents generated by the node, another for Contents requested or passing through it), or to set a minimum time limit until a generated Content by itself can be considered in the replacement process. This would lead to an attempt not to delete Contents early, thus increasing the cache and the network performance.

5. Conclusions and future work

The work done on the in-network caching scheme led to a caching proposal based on the use of a cache admission policy, entitled as *ctxCAP*, which takes into account context-related parameters such as Content popularity, Content freshness, the proximity to the consumers that requested the Content and the Content provider mobility type. Each node integrates *ctxCAP*, which, in combination with a replacement policy, implements a complete caching strategy.

By simulating and evaluating several scenarios, which incorporate real mobility traces from a large-scale vehicular scenario, it was possible to evaluate the performance of the *ctxCAP* in comparison with the *LCE* strategy, the one adopted by the NDN, as well as two replacement strategies: *PFIFO* and *LRU*. The results show that *ctxCAP* improves the cache hits, cache evictions, and the percentage of satisfied requests considering intermediate nodes and consumers percentage by a large margin. The use of a more sophisticated caching scheme has strong benefits in a mobile environment, especially in the users' satisfaction.

However, the work can be improved in several ways, such as the integration of more parameters and automation of the threshold value used by the cache admission policy algorithm, and also the differentiation of the produced Contents in the caching process.

CRediT authorship contribution statement

Luís Leira: Conceptualization, Methodology, Validation, Data curation, Writing – original draft, Visualization. **Miguel Luís:** Conceptualization, Validation, Writing – review & editing, Visualization, Funding acquisition. **Susana Sargento:** Conceptualization, Validation, Writing – review & editing, Visualization, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020, Regional Operational Program of Lisbon (FEDER), and Foundation for Science and Technology, Portugal, through project InfoCent-IoT - Efficient information centric networks for IoT infrastructures (POCI-01-0145-FEDER-030433).

References

[1] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660, http://dx.doi.org/10.1016/j.future.2013.01.010.

[2] B. Nour, K. Sharif, F. Li, S. Biswas, H. Moungla, M. Guizani, Y. Wang, A survey of Internet of Things communication using ICN: A use case perspective, Comput. Commun. 142–143 (2019) 95–123, http://dx.doi.org/10.1016/j.comcom.2019.05.010.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, Internet of things for smart cities, IEEE Internet Things J. 1 (1) (2014) 22–32, http://dx.doi.org/10.1109/JIOT.2014.2306328.

[4] E.T. Chen, The internet of things: Opportunities, issues, and challenges, in: The Internet of Things in the Modern Business Environment, IGI Global, 2017, pp. 167–187, Ch. 10.

[5] J. Pan, S. Paul, R. Jain, A survey of the research on future internet architectures, IEEE Commun. Mag. 49 (7) (2011) 26–36, http://dx.doi.org/10.1109/MCOM.2011.5936152.

[6] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R.L. Aguiar, A.V. Vasilakos, Information-centric networking for the internet of things: challenges and opportunities, IEEE Netw. 30 (2) (2016) 92–100, http://dx.doi.org/10.1109/MNET.2016.7437030.

[7] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Commun. Mag. 50 (7) (2012) 26–36, http://dx.doi.org/10.1109/MCOM.2012.6231276.

[8] M. Meisel, V.L. Pappas, L. Zhang, Listen first, broadcast later: Topology-agnostic forwarding under high dynamics, in: Annual Conference of International Technology Alliance in Network and Information Science, 2010.

[9] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, S. AlAhmadi, Cache freshness in named data networking for the internet of things, Comput. J. 61 (10) (2018) 1496–1511, http://dx.doi.org/10.1093/comjnl/bxy005.

[10] M. Amadeo, C. Campolo, G. Ruggeri, G. Lia, A. Molinaro, Caching transient contents in vehicular named data networking: A performance analysis, Sensors 20 (7) (2020).

[11] M. Zhang, H. Luo, H. Zhang, A survey of caching mechanisms in information-centric networking, IEEE Commun. Surv. Tutor. 17 (3) (2015) 1473–1499, http://dx.doi.org/10.1109/COMST.2015.2420097.

[12] A. Ioannou, S. Weber, A survey of caching policies and forwarding mechanisms in information-centric networking, IEEE Commun. Surv. Tutor. 18 (4) (2016) 2847–2886, http://dx.doi.org/10.1109/COMST.2016.2565541.

[13] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, SIGCOMM Comput. Commun. Rev. 44 (3) (2014) 66–73, http://dx.doi.org/10.1145/2656877.2656887.

[14] S.K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, S. Shenker, Less pain, most of the gain: Incrementally deployable ICN, in: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, in: SIGCOMM '13, ACM, New York, NY, USA, 2013, pp. 147–158, http://dx.doi.org/10.1145/2486001.2486023.

[15] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, Cache coherence in machine-to-machine information centric networks, in: 2015 IEEE 40th Conference on Local Computer Networks (LCN), 2015, pp. 430–433, http://dx.doi.org/10.1109/LCN.2015.7366345.

[16] L. Saino, I. Psaras, G. Pavlou, Icarus: A caching simulator for information centric networking (ICN), in: Proceedings of the 7th International ICST Conference on Simulation Tools and Techniques, in: SIMUTools '14, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, 2014, pp. 66–75, http://dx.doi.org/10.4108/icst.simutools.2014.254630.

[17] W.K. Chai, D. He, I. Psaras, G. Pavlou, Cache "less for more" in information-centric networks, in: Proceedings of the 11th International IFIP TC 6 Conference on Networking - Volume Part I, in: IFIP'12, Springer-Verlag, Berlin, Heidelberg, 2012, pp. 27–40, http://dx.doi.org/10.1007/978-3-642-30045-5_3.

[18] I. Psaras, W.K. Chai, G. Pavlou, Probabilistic in-network caching for information-centric networks, in: Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking, in: ICN '12, ACM, New York, NY, USA, 2012, pp. 55–60, http://dx.doi.org/10.1145/2342488.2342501.

[19] S. Podlipnig, L. Böszörmenyi, A survey of web cache replacement strategies, ACM Comput. Surv. 35 (4) (2003) 374–398, http://dx.doi.org/10.1145/954339.954341.

[20] M. Bilal, S. Kang, A cache management scheme for efficient content eviction and replication in cache networks, IEEE Access 5 (2017) 1692–1701, http://dx.doi.org/10.1109/ACCESS.2017.2669344.

[21] B. Panigrahi, S. Shailendra, H.K. Rath, A. Simha, Universal caching model and Markov-based cache analysis for information centric networks, in: 2014 IEEE International Conference on Advanced Networks and Telecommuncations Systems (ANTS), 2014, pp. 1–6, http://dx.doi.org/10.1109/ANTS.2014.7057262.

[22] B. Banerjee, A. Kulkarni, A. Seetharam, Greedy caching: An optimized content placement strategy for information-centric networks, Comput. Netw. 140 (2018) 78–91, http://dx.doi.org/10.1016/j.comnet.2018.05.001.

[23] J. Pfender, A. Valera, W.K.G. Seah, Performance comparison of caching strategies for information-centric IoT, in: Proceedings of the 5th ACM Conference on Information-Centric Networking, in: ICN '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 43–53, http://dx.doi.org/10.1145/3267955.3267966.

[24] M.A. Hail, M. Amadeo, A. Molinaro, S. Fischer, Caching in named data networking for the wireless internet of things, in: 2015 International Conference on Recent Advances in Internet of Things (RIoT), 2015, pp. 1–6, http://dx.doi.org/10.1109/RIOT.2015.7104902.

[25] S. Vural, P. Navaratnam, N. Wang, C. Wang, L. Dong, R. Tafazolli, In-network caching of Internet-of-Things data, in: 2014 IEEE International Conference on Communications (ICC), 2014, pp. 3185–3190, http://dx.doi.org/10.1109/ICC.2014.6883811.

[26] M. Hail, M. Amadeo, A. Molinaro, S. Fischer, On the Performance of Caching and Forwarding in Information-Centric Networking for the IoT, Vol. 9071, 2015, http://dx.doi.org/10.1007/978-3-319-22572-2_23.

[27] B. Nour, K. Sharif, F. Li, H. Moungla, A.E. Kamal, H. Afifi, NCP: A near ICN cache placement scheme for IoT-based traffic class, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6, http://dx.doi.org/10.1109/GLOCOM.2018.8647629.

[28] J. Pfender, A. Valera, W.K. Seah, Easy as ABC: A lightweight centrality-based caching strategy for information-centric IoT, in: Proceedings of the 6th ACM Conference on Information-Centric Networking, in: ICN '19, ACM, New York, NY, USA, 2019, pp. 100–111, http://dx.doi.org/10.1145/3357150.3357405.

[29] M. Naeem, R. Ali, B.-S. Kim, S. Awang Nor, S. Hassan, A periodic caching strategy solution for the smart city in information-centric internet of things, Sustainability 10 (2018) 2576, http://dx.doi.org/10.3390/su10072576.

[30] A. Gomes, T. Braun, Feasibility of information-centric networking integration into lte mobile networks, in: Proceedings of the 30th Annual ACM Symposium on Applied Computing, in: SAC '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 627–633, http://dx.doi.org/10.1145/2695664.2695790.

[31] S.-E. Elayoubi, J. Roberts, Performance and cost effectiveness of caching in mobile access networks, in: Proceedings of the 2nd ACM Conference on Information-Centric Networking, in: ACM-ICN '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 79–88, http://dx.doi.org/10.1145/2810156.2810168.

[32] G. Zheng, V. Friderikos, Optimal proactive cache management in mobile networks, in: 2016 IEEE International Conference on Communications (ICC), 2016, pp. 1–6, http://dx.doi.org/10.1109/ICC.2016.7511464.

[33] Z. Zhang, C.-H. Lung, I. Lambadaris, M. St-Hilaire, When 5G meets ICN: An ICN-based caching approach for mobile video in 5G networks, Comput. Commun. 118 (2018) 81–92, http://dx.doi.org/10.1016/j.comcom.2017.10.002.

[34] C. Ge, N. Wang, S. Skillman, G. Foster, Y. Cao, QoE-driven DASH video caching and adaptation at 5G mobile edge, in: Proceedings of the 3rd ACM Conference on Information-Centric Networking, in: ACM-ICN '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 237–242, http://dx.doi.org/10.1145/2984356.2988522.

[35] F.M. Modesto, A. Boukerche, An analysis of caching in information-centric vehicular networks, in: 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–6, http://dx.doi.org/10.1109/ICC.2017.7997019.

[36] D. Grewe, M. Wagner, H. Frey, PeRCeIVE: Proactive caching in ICN-based VANETs, in: 2016 IEEE Vehicular Networking Conference (VNC), 2016, pp. 1–8, http://dx.doi.org/10.1109/VNC.2016.7835962.

[37] Z. Zhang, C.-H. Lung, M. St-Hilaire, I. Lambadaris, Smart proactive caching: Empower the video delivery for autonomous vehicles in ICN-based networks, IEEE Trans. Veh. Technol. 69 (7) (2020) 7955–7965, http://dx.doi.org/10.1109/TVT.2020.2994181.

[38] H. Zhang, L. Dai, Mobility prediction: A survey on state-of-the-art schemes and future applications, IEEE Access 7 (2019) 802–822, http://dx.doi.org/10.1109/ACCESS.2018.2885821.

[39] L. Irio, A. Ip, R. Oliveira, M. Luís, An adaptive learning-based approach for vehicle mobility prediction, IEEE Access 9 (2021) 13671–13682, http://dx.doi.org/10.1109/ACCESS.2021.3052071.

[40] A. Cardote, F. Neves, S. Sargento, P. Steenkiste, A statistical channel model for realistic simulation in VANET, in: 2012 IEEE Vehicular Networking Conference (VNC), 2012, pp. 48–55.