Reliable data delivery in ICN–IoT environments

Eleonora Borgia [*], Raffaele Bruno, Andrea Passarella

Institute for Informatics and Telematics, National Research Council. Via G. Moruzzi 1, 56124 Pisa, Italy

ABSTRACT

In an IoT environment, which is characterized by a multitude of interconnected smart devices with sensing and computational capabilities, many applications are (i) content-based, that is, they are only interested in finding a given type of content rather than the location of data, and (ii) contextualized, that is, the content is generated and consumed in the proximity of the user. In this context, the Information-Centric Networking (ICN) paradigm is an appealing model for efficiently retrieving application data, and the service decentralization towards the network edge helps to reduce the core network load being the data produced by IoT devices mainly confined in the area where they are generated. MobCCN is an ICN-based data delivery protocol that we designed for operating efficiently in such context (Borgia et al., 2016, 2018), where static and mobile IoT devices are enriched with ICN functions. Specifically, MobCCN leverages an efficient routing and forwarding protocol, exploiting opportunistic contacts among IoT mobile devices, to fill the Forwarding Interest Base (FIB) tables so as to correctly forward Interest packets towards the intended data producers. In this paper, we aim to enhance the reliability of MobCCN by exploring different retransmissions mechanisms, such as retransmissions based on number of duplicate Interests that are received for the same requested content, periodic retransmissions, single path versus disjoint multi-path forwarding, hysteresis mechanism and combinations of them. Extensive simulation results show that, among the analysed MobCCN variants, the one that implements both periodic retransmissions and a hysteresis-based retransmission process ensures the highest delivery rates (up to 95%) and the shortest network path, with a very limited traffic overhead due to retransmissions.

1. Introduction

The *Internet of Things (IoT)* refers to the paradigm where a massive number of inter-connected devices with sensing and computational capabilities, uniquely addressable, forms a dynamic network to communicate with each other or with the Internet without human intervention [1]. IoT devices are extremely heterogeneous in terms of resource capabilities, lifespan and communication technologies. On the one end of the spectrum, there are static, tiny and battery-powered devices with limited computational and memory capabilities, which are embedded with sensors and actuators. These IoT devices are typically used to collect and exchange sensed data about the state of physical objects, processes and environments, and to react to this information by performing appropriate control actions. On the other end of the spectrum, the physical environment is also sprinkled with more powerful IoT devices, e.g., smartphones carried by mobile users, or vehicle-based sensing platforms, which can produce, collect, request, consume data through context-aware

applications. Previous research studies have proposed to leverage these less resource-constrained mobile devices to act as "data mules" so as to facilitate the collection and distribution to other nodes of the data generated by the tiny sensor nodes [2–4]. Recently, this architectural design has gained further popularity thanks to new 4G/5G capabilities, and in particular D2D communications and Proximity Services (ProSe), which provide the enabling mechanisms to discover other mobile devices in close proximity and to communicate with them directly [5,6]. Motivated by these considerations, we envision a scenario where at the edge of the network there are regions where IoT data is produced by local static IoT devices, and this data is collected and distributed to other devices in the overall area by leveraging an intermediate layer of mobile users' devices, as shown in Fig. 1. Typical application scenarios for this framework can be identified in the smart-city context. For instance, localized information about discounts or coupon deals from nearest shops in a mall can be forwarded using D2D communications between users' personal devices to the people walking in the area. Similarly, images captured by camera-based sensors that are installed in city hot spots can forwarded to the interested users without using expensive cellular communications. Note that these applications are typically delay tolerant and not critical, namely minimal latency

is not a critical factor. Furthermore, the intrinsic data redundancy (e.g., multiple cameras taking similar images of a scene) makes acceptable to loose some of the transmitted messages.

From the above discussion, it should be clear that this study specifically targets the emerging category of *content-centric* IoT applications rather than host-centric ones. In such type of applications, the goal of the communication process is to search for a specific content (mainly generated in the proximity of the user), and not to identify the location of the device that originated or currently stores that content. It is important to remind that the *Information-Centric Networking* (ICN) architecture [7] was originally proposed to support content-centric applications in the Internet, such as file sharing and media streaming. However, the ICN paradigm has been also proposed as an alternative networking architecture for IoT [8,9]. The basic ICN design principle is to adopt content naming, meaning that each content is assigned a unique and location-independent name. The content retrieval process then follows a receiver-driven approach, as content segments (or chunks) are exchanged using a request/response model. Additional appealing ICN features are in-networking caching, request aggregation, mobility support, and content security [7].

As the integration of ICN mechanisms into IoT environments has a strong potential to fulfil the requirements of content-centric IoT applications, in this work we investigate how to implement an ICN-based framework in the scenario illustrated in Fig. 1, to provide reliable and scalable IoT data collection and distribution by leveraging users' mobile devices. In our previous work we have already proposed MobCCN, a routing and forwarding protocol that is compliant with ICN paradigms (namely CCN [10] and NDN [11]), and leverages D2D communications to allows users' mobile devices to opportunistically exchanged data collected by nearby static IoT devices. To perform precise forwarding decisions, MobCCN exploits the concept of utility of a node towards a given content, i.e., a measurement of how frequently a node meets other nodes that store the requested content in their local caches. These utility values are exchanged during opportunistic contacts among mobile nodes, to create a dynamic gradient-based content-dissemination graph used to redirect content requests and retrieve the content efficiently. We presented the main idea of MobCCN in [12] and its performance in [13], where we have proved that, although additional signalling traffic is needed to build more efficient forwarding tables by computing and exchanging the utility value for each type of content available in the network, this approach ensures a more precise detection of either the relevant content producers or the nodes that cache the relevant content. This yields a remarkable reduction of total network traffic and cache usage when compared with an epidemic-based routing protocol with a small degradation of the delivery rate in most cases, and an increase of end-to-end delays.

The retransmission-based loss recovery mechanisms typically used by CCN, which are inherited in MobCCN, are known to be not very efficient in lossy or mobile networks [14]. The goal of this study is to shed some light on the performance improvements that could be achieved if more sophisticated retransmission mechanisms are used in MobCCN, and the consequential trade-offs between increased reliability (i.e., higher delivery rate), and delays or network traffic overheads. Specifically, the standard CCN retransmission scheme simply retransmits pending Interest packets periodically or when a duplicate one is received, using the reverse path traversed by the interest packet. In this work we explore alternative retransmission approaches. Specifically, we consider schemes that leverage *path diversity* (i.e., the packet loss processes are expected to operate independently for different paths) by sending retransmissions over a different path than the reverse path. Note that multi-path routing is also beneficial to
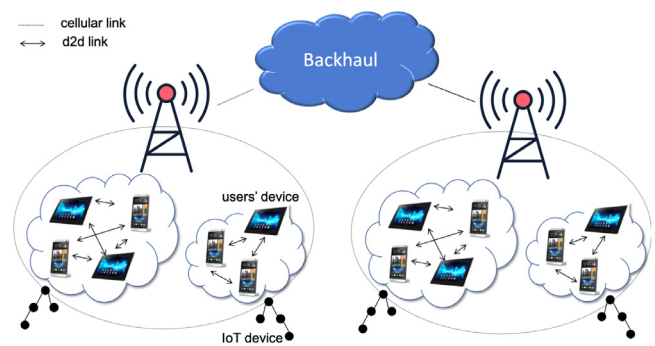

Fig. 1. Illustrative example of the reference network scenario considered in this study.

distribute more evenly the traffic load due to retransmissions. In addition, we also investigate adaptive scheme to set up the periodic retransmission timer, which leverages *utility hysteresis* to control the retransmission frequency. We have integrated the proposed retransmission strategies into the MobCCN prototype, and evaluated the advantages and disadvantages of each of them under a variety of scenarios using the OMNeT++ network simulator.[1] Simulation results show that the hysteresis-based approach, called MobCCN_AH, achieves the best performance, with delivery rates in the range of [85–95]%, about 15%–20% higher than the ones obtained by basic MobCCN, when sufficient bandwidth is available for the retransmissions. On the contrary, in bandwidth-limited scenarios the performance of the different retransmission policies tends to converge.

The rest of the paper is organized as follows. Section 2 overviews the most related research work. Section 3 briefly presents the basic MobCCN protocol. In Section 4 we describe the proposed retransmission strategies. Section 5 introduces the simulation setup. In Section 6 we present the comprehensive performance comparison. Finally, Section 7 draws the main conclusions of the paper.

2. Related work

In the recent years, an increasing number of research studies have been dedicated to investigate critical aspects and challenges for the integration of ICN mechanisms into IoT networks [15–18].

One line of research that is well investigated focuses on the problem of efficient data retrieval, which consists in specifying the routing and forwarding processes. Proactive link-state routing solutions, where nodes periodically exchange information to fill the forwarding tables, have been typically proposed for network scenarios with static nodes (see for example OSPFN [19] or NLSR [20]). On the contrary, routing protocols for ICN-IoT environments with mobile nodes are less investigated. DABBER [21], for instance, supports opportunistic communications. It relies on node context information such as availability and centrality of adjacent nodes, as well as of the availability of different data sources to drive forwarding of Interests. When an underlying routing protocol populating FIBs is missing, the forwarding plane acts as a reactive routing protocol by exploiting on-demand or controlled flooding to decide the best next-hop forwarder [22]. To further refine forwarding decisions, some solutions use the context of the requested content and the tolerance to inaccuracies in data defined for the application requesting it [23], or the network performance measurements [24], or a mix of context information and optimal selection of the forwarding times [25].

---

1 http://www.omnetpp.org/.

As an alternative to name-based flooding, location-based solutions that exploit the geographic coordinates of nodes are also investigated, primarily in the context of vehicular networks [26] and recently in IoT systems [27]. To improve the scalability of forwarding tables, solutions based on bloom filters are proposed in [28–30]. There exists also a class of routing protocols for ICN-IoT environments specifically designed for push-based traffic, such as that produced in case of alarms or status changes. These schemes allow data producers to distribute data to subscribed consumers even without explicit and continuous requests [31,32].

The mobility of nodes increases the complexity level of designing an efficient data retrieval. Mobile devices can be content consumers, or content producers, but they can also play the role of consumers and producers at the same time. Basically, ICN networks support consumer mobility by design. In contrast with IP-networks, where an update of the address information is needed each time a mobile node changes the network as happens in Mobile IP, the native mobility ICN support consists just in reissuing the content request for those lost contents. However, such solution is not always sufficient, especially in the IoT context, but more complex schemes are required such as rendezvous-based schemes [33], or synchronization of the subscription table during consumers' movements [34]. On the contrary, providers' mobility is more challenging because it is necessary to maintain track of the providers' locations in order to guarantee the session maintenance. Among the several classes of proposed solutions, multi-path based schemes simultaneously transmit multiple copies of content requests to potential locations where the providers may move [35]. The use of caching also contributes to mitigate movements of both consumers [36] and producers [37], as well as improving system performance, especially in terms of experienced delay.

To the best of our knowledge, the design of reliability mechanisms to improve the robustness of an ICN-based data delivery process has not been sufficiently investigated. Most of the existing solutions assign the execution of the proposed reliability mechanisms to the consumers, which are in charge of simply reissuing Interest packets for those Data packets that were never received. Retransmissions are usually triggered by the application layer when a specific retransmission timeout (RTO) for a content expires. Almost all solutions focus exclusively on the estimation of such timeout, for example by including a mechanism similar to TCP EWMA, or by excluding any RTO updates in case of Data reception due to retransmissions [38,39], or estimating different timeout values for each content producer [40]. None of the aforementioned studies attempts to improve the retransmission mechanisms at the network level nor do they explore the effectiveness of retransmission mechanisms that are not based on timeouts, which is one of the main objectives of this work. In our previous paper [13] we have already proved that, when applying the standard retransmission mechanism adopted by the vanilla ICN, there are cases in which the delivery rate does not improve but only a traffic overhead increase is observed. An additional issue that we have also observed is the occurrence of network loops for Interest packets. Thus, further investigations are needed to improve the network-level retransmission mechanism in ICN systems. A recent and orthogonal direction is the use of network coding in ICN networks to increase the robustness to content delivery [41]. Some solutions are presented in [42,43]. However, both works assume communications over wired networks to which devices with multiple (wireless) interfaces try to access, therefore a very different environment with the one considered in this paper.

Finally, several prototypes of ICN-based solutions for IoT networks have been developed using software platforms and
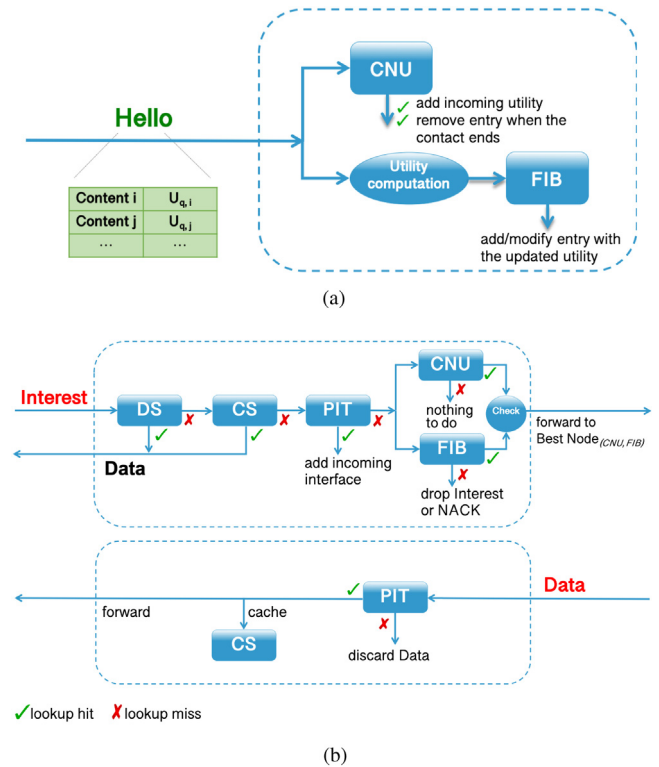

Fig. 2. MobCCN protocol: routing phase (2(a)), and forwarding phase (2(b)).

operating systems that were specifically designed for resource-constrained devices, such as CCN-lite [44], RIoT [45], and Contiki [46]. Large-scale real-world test-beds have been used to analyse the performances of these prototypes from different perspectives, such as resource (e.g., memory, cpu, energy) consumption [47,48], comparison between single-hop and multi-hop network typologies [47,49], and comparison of ICN-based architectures against IP-based ones [47,50]. However, most of these experimental studies consider only static network scenarios.

3. Background: the MobCCN protocol

For the sake of clarity, in this section we overview the general design principles and the basic mechanisms of MobCCN protocol. We refer to [12] and to [13] for a more detailed description.

As explained in Section 1, the design rationale of MobCCN is to use an ICN-based communication paradigm to build a data management layer that is made of mobile devices using D2D communications to exchange messages. This data management platform is then used to collect and distribute data generated by nearby sensor nodes and other mobile devices. MobCCN inherits all the basic mechanisms and concepts of ICN solutions, and in particular the ones of the CCN/NDN architecture. Specifically, in MobCCN, Interest packets are used to request contents, Data packets carry the requested contents, the Content Store (CS) is used for storing the content received by a node, the Pending Interest Table (PIT) takes note of Interest packets arrived at one node but not yet satisfied, and the Forwarding Interest Table (FIB) serves as the routing table of Interest packets, similarly to the packet routing table of a legacy Internet node. In MobCCN a data store, called DS, is used not only to maintain the data that is produced by the node itself, but also to store the contents that are retrieved by the sensor nodes that are encountered. The key concept behind MobCCN routing process is the *utility value* of a node p for a specific content i ($U_{p,i}$), which is a function of both

the frequency node p encounters the node that generated content i (*direct utility* component) and the frequency node p encounters other nodes that have earlier "bumped" into that content, i.e., which have non-null utility towards content i (*indirect utility* component). Formally, the direct utility of a node p towards a content i is defined as:

$$U_{p,i}^{(d)} = \frac{1}{ICT(p, i)} \qquad (1)$$

where $ICT(p, i)$ is the inter-contact time between p and any node that generated content i. The indirect utility is defined as:

$$U_{p,q,i}^{(ind)} = \frac{1}{\frac{1}{U_{q,i}} + ICT(p, q)} \, , \qquad (2)$$

where q is any node encountered by node p, which does not store the content i, $U_{q,i}$ is the current utility of node q towards content i, and $ICT(p, q)$ is the inter-contact time between p and q. Direct and indirect utilities are then merged together to provide the global utility value of the node p for the content i as follows:

$$U_{p,i} = \max_{j \in \mathcal{N}_p} \left\{ U_{p,i}^{(d)}, U_{p,j,i}^{(ind)} \right\} \, , \qquad (3)$$

where $\mathcal{N}_p$ is the set of nodes that p met. The utility value $U_{p,i}$, computed and updated at each contact by exchanging *Hello* packets, is used to create a *utility-gradient-based information-centric graph* during the routing phase, and to forward the Interest packets during the forwarding phase.

MobCCN uses the *non-permanent* in-network caching policy of ICN, namely each node stores into its CS a copy of the content items it receives, but only for the amount of time that is necessary to serve all pending requests in the PIT. Therefore, in MobCCN a node does not announce a direct utility value for the contents that are added to its CS as they are only temporarily stored and the time they are maintained into the case is variable. In fact, using these copies of the content in the direct utility computation may lead to inconsistent routing decisions in other nodes. During the routing phase (see Fig. 2(a)) when two nodes meet, each of them broadcasts a Hello packet to advertise the utilities of the contents it knows, as well as the identifiers of the nodes with the highest utility values for those contents. Then, for each content i advertised in the received Hello packet the tagged node p: *(1)* creates a temporary entry in the Current Neighbours Utilities (CNU) table for content i by storing the utility received by the neighbour[2], *(2)* computes the new utility value using Eq. (1) or Eq. (2) depending on whether the encountered node is a producer or not for content i, and *(3)* creates a new (or updates the corresponding) entry in the FIB table for content i by storing the interface (or "face"[3]) from which the Hello packet came and the utility value computed earlier. In this way, the FIB table is populated with the information required to build a utility-gradient-based information-centric graph for all the contents available in the network.

During the forwarding phase (see Fig. 2(b)), when a node p receives an Interest packet for the content i, first it checks if the requested content is stored locally in CS and, if the case, it forwards the Data packet to the node from which the request arrived. If the content is not stored in the CS, it generates or updates the corresponding PIT entry. These two operations are performed exactly as in the original CCN. However, unlike CCN, the utility-gradient-based information-centric graph created during the routing phase is used to forward the Interest packet. Specifically, node p identifies the node with the highest utility towards content i, searching both its current neighbours and other

---

[2] This entry is then removed upon the expiration of contact.

[3] As we adopted the same CCN terminology, we use node/face/interface interchangeably in the text.
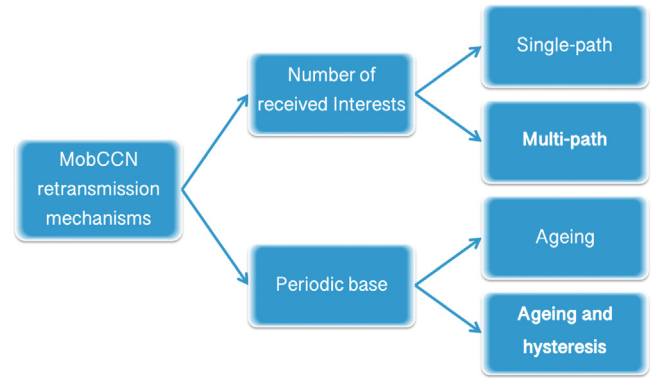

Fig. 3. Schematic representation of the MobCCN retransmission mechanisms.

nodes from which it collected routing information in previous contacts. More precisely, MobCCN queries the CNU table to find the neighbour with the highest utility towards content i; and the FIB table to find the best candidate forwarders for that Interest packet. Only if the two nodes are the same, the Interest packet is forwarded immediately. Otherwise, the Interest packet is stored at node p, which repeats this selection procedure whenever the set of current neighbours changes. Forwarding the Interest packet following opportunistically the positive gradient ensures that the required content is discovered efficiently. Once the content is found, it is encapsulated in a Data packet and forwarded to the requester following the standard CCN mechanism that exploits the reverse-path established at each node.

4. MobCCN reliability enhancements

In the following we describe the retransmission mechanisms that we added to MobCCN in order to improve its reliability. Specifically, Section 4.1 describes the two mechanisms that leverage the number of received Interests to trigger the retransmission of a content, while Section 4.2 focuses on mechanisms based on timer expiration. It is worth noting that one of the two mechanisms of each category is inherited from the ones adopted by the vanilla CCN, and they are used as benchmarks for those that are proposed by this study, and which represent the original contribution of this paper. For the sake of presentation clarity, Fig. 3 shows a schematic classification of the retransmission mechanisms that we discuss in the following, where we mark our original contributions in bold.

4.1. Retransmission mechanisms based on the number of received requests

In the following sub-section we describe the two solutions that differ in the way they route the pending Interest packets.

4.1.1. Single-path mechanism

First of all, we analyse the standard retransmission mechanism adopted by the vanilla CCN in which a retransmission occurs according to the number of the Interest packets for the same content received by a node. Algorithm 1 describes the operations of this scheme executed when Interest packets arrive at one node. More specifically, if a node p receives an Interest packet that it cannot satisfy (i.e., the content is not stored in the CS), it generates a PIT entry and forwards the Interest to the node with the highest utility for that content among those in FIB and CNU tables (line 10–21). If p receives a second Interest packet

Algorithm 1 MobCCN_R1: processing of Interest packets received by node p

---
Require: E_p ▷ Set of current neighbours of node p

1: $i \leftarrow Interest.\text{Name}()$
2: **if** Data $\leftarrow$ CS.Find(i) **then**
3:    **return** Data
4: **end if**
5: **if** PIT.Find(i) **then**
6:    PIT.AddFace(i, q)
7:    **if** PIT.numberOfFace(i) $> ReTX_{threshold}$ **then**   ▷ old Interest, nevertheless
8:   propagated with R1
9:      R1(i, Interest)
10:   **end if**
11: **else**
12:   PIT.CreateFace(i, q)
13:   **if** FIB.Find(i) **then**
14:     find the node j with the highest utility for content type i in FIB and
15:     CNU tables
16:     **if** $j \in E_p$ **then**
17:       forward Interest packet to node j
18:     **else**
19:       store Interest packet waiting for the next contact
20:     **end if**
21:   **else**
22:     drop Interest packet
23:   **end if**
24: **end if**

25: **procedure** R1($i, Interest$)
26:   **if** FIB.Find(i) **then**
27:     find the node j with the highest utility for content type i in FIB and
28:     CNU tables
29:     **if** $j \in E_p$ **then**
30:       forward Interest packet to node j
31:     **else**
32:       wait for the next contact
33:     **end if**
34:   **else**
35:     drop Interest packet
36:   **end if**
37: **end procedure**

Algorithm 2 MobCCN_R2: processing of Interest packets received by node p

---
Require: E_p ▷ Set of current neighbours of node p

1: $i \leftarrow Interest.\text{Name}()$
2: **if** Data $\leftarrow$ CS.Find(i) **then**
3:    **return** Data
4: **end if**
5: **if** PIT.Find(i) **then**
6:    PIT.AddFace(i, q)
7:    **if** PIT.numberOfFace(i) $> ReTX_{threshold}$ **then**   ▷ old Interest, nevertheless
8:   propagated with R2
9:      R2(i, Interest)
10:   **end if**
11: **else**
12:   PIT.CreateFace(i, q)
13:   **if** FIB.Find(i) **then**
14:     find the node j with the highest utility for content type i in FIB and
15:     CNU tables
16:     **if** $j \in E_p$ **then**
17:       forward Interest packet to node j
18:     **else**
19:       store Interest packet waiting for the next contact
20:     **end if**
21:   **else**
22:     drop Interest packet
23:   **end if**
24: **end if**

25: **procedure** R2($i, Interest$)
26:   **if** FIB.Find(i) **then**
27:     find the node j with the second highest utility for content type i in FIB
28:     and CNU tables
29:     **if** $j \in E_p$ **then**
30:       forward Interest packet to node j
31:     **else**
32:       wait for the next contact
33:     **end if**
34:   **else**
35:     drop Interest packet
36:   **end if**
37: **end procedure**

for the same content i of which the corresponding Data message is still missing, it adds the incoming "face" to the corresponding PIT entry (line 6). If the number of received Interests for the same content exceeds a certain retransmission threshold (i.e., ReTX$_{threshold}$ at line 7), p retransmits the first received Interest towards the node with the highest utility for that content (line 23–34). It is important to observe that if the selected forwarder is not currently in proximity of node p (i.e., the forwarder is not a neighbour of node p), the retransmission is delayed till the next contact. Intuitively, if the $ReTX_{threshold}$ is set equal to 0 then each duplicate reception of an Interest packet triggers a retransmission. Obviously, the smaller the $ReTX_{threshold}$ and the higher the retransmission traffic that is generated. Nevertheless, we expect that a trade-off exists between the additional network congestion caused by the transmitted packets and the gain in number of data requests that can be successfully satisfied. In the rest of the paper, we refer to this solution as "*MobCCN_R1*".

Algorithm 3 MobCCN_A: procedure at node p at the expiration of the ageing timeout T$_{ageing}$

Require: E_p ▷ Set of current neighbours of node p

1: **for each** $e \in PIT$ **do**
2:   **if** **not** PIT.Expired(e) **then**
3:     $i \leftarrow$ PIT.Name(e)
4:     **if** FIB.Find(i) **then**
5:       find the node j with the highest utility for content type i in FIB and
6:       CNU tables
7:       **if** $j \in E_p$ **then**
8:         forward Interest packet to node j
9:       **else**
10:        wait for the next contact
11:       **end if**
12:     **else**
13:       drop Interest packet
14:     **end if**
15:   **else**
16:     drop Interest packet
17:     PIT.remove(e)
18:   **end if**
19: **end for**

4.1.2. Multi-path mechanism

We introduce a protocol variant that adopts the same rule of MobCCN_R1 to trigger a packet retransmission (i.e., the number of duplicated Interest packets goes above a defined threshold), but also leverages path diversity when retransmitting Interests. The idea here is that, instead of using the same path to retransmit Interests, the node explores a second path to find the content. To achieve this, the Interest to be retransmitted is sent to the second best forwarder. The protocol works as follows (see Algorithm 2). As before, if multiple Interest packets for the same content arrive at one node, the node forwards only the first Interest to the node with the highest utility it knows (line 11–21), storing all the other "faces" in the corresponding PIT entry (line 6). However, if a retransmission event is triggered (line 7), the node retransmits the first Interest packet it has received to the second best forwarder, i.e., to the node with the second highest utility it knows (line 23–34). In this way, a second path is explored, hopefully increasing the probability of finding the content. What we expect to obtain with this retransmission approach is an improvement of the delivery rate but at the cost of a higher end-to-end delay and hop count due to the presence of non-optimal paths. In the following, we call this alternative solution "*MobCCN_R2*".

4.2. Retransmission mechanisms based on timeout expiration

In the following sub-section we describe the two solutions based on periodic retransmission of the pending Interest packets.

4.2.1. Ageing mechanism

The rationale behind this standard retransmission mechanism of the vanilla CCN that is adopted by MobCCN is to activate the retransmission mechanism according to the expiration of a specified timeout parameter. Basically, this version behaves in the same way of the MobCCN_basic for what concerns the processing of Interest packets received by nodes as depicted in Fig. 2(b), but the retransmission of pending Interests is executed periodically (see Algorithm 3). Specifically, whenever the *ageing timeout* (T_{ageing}) expires, each node p checks its PIT table looking for still valid entries that can also be retransmitted (line 2). For those entries matching the above criteria, the retransmission of the Interest packet towards the node with highest utility occurs (line 3–14). Because of the timeout, the network has time to possibly reconfigure, and thus improve the accuracy of the ranking of forwarders towards the requested content. If there are old Interests, the corresponding entries are deleted from the PIT so as not become stale and be held indefinitely (line 16). What we want to assess here is the potential benefit of a periodic retransmission on protocol performance. Obviously, an increase of the traffic sent with respect to the MobCCN_basic is expected, but this can be limited by tuning conveniently the ageing timeout. This solution is referred as "*MobCCN_A*".

4.2.2. Ageing and hysteresis mechanisms

In this further version we use a combination of ageing and hysteresis mechanisms. Specifically, we enrich MobCCN_A with a hysteresis mechanism aimed at improving the selection of the node to which to forward the Interest packets. Indeed, the encountered node is used as forwarded only if it improves by at least a threshold, called *Hyst* value, the utility value of the node p of forwarding the Interest to the newly encountered node. Algorithms 4 and 5 provide the description of the steps executed by node p at the reception of an Interest packet and at the expiration of the ageing timeout, respectively. As it is apparent, the differences consist only in the condition that has to be satisfied for the transmission/retransmission of an Interest packet. Taking Algorithm 4 as example, first the node j with the highest utility value in FIB and CNU is identified. If j is a neighbour of p, and the difference between the utility values of node j and node p is not less than a given threshold (determined by the hysteresis parameter), node p forwards the Interest packet to the encountered node j (lines 13–16), otherwise the Interest packet is stored waiting for the next contact (lines 17–18). The same condition holds at lines 10–18 of Algorithm 5. The idea here is to verify if using a more conservative forwarding strategy improves the protocol performance. Postponing the Interest forwarding until a node whose utility value is significantly greater is found, is expected to guarantee a faster Interest convergence towards the content provider with a consequent decrease in paths length. In the rest of the paper this solution is referred to as "*MobCCN_AH*".

5. Simulation setup

To evaluate the performance of the proposed retransmission mechanisms, we extended the prototype of the MobCCN protocol described in [12] with the enhancements described in Section 4. We remind that MobCCN was implemented as an extension of CCN-lite [44], a lightweight open source CCN implementation

Algorithm 4 MobCCN_AH: processing of Interest packets received by node p

Require: E_p ▷ Set of current neighbours of node p

```
1:  i ← Interest.Name()
2:  if Data ← CS.Find(i) then
3:      return Data
4:  end if
5:  if PIT.Find(i) then
6:      PIT.AddFace(i, q)
7:  else
8:      PIT.CreateFace(i, q)
9:      if FIB.Find(i) then
10:         find the node j with the highest utility for content type i in FIB and
11:         CNU tables
12:         if j ∈ E_p  and  U_{j,i} > U_{p,i} * (1 + Hyst/100) then
13:             forward Interest packet to node j
14:         else
15:             store Interest packet waiting for the next contact
16:         end if
17:     else
18:         drop Interest packet
19:     end if
20: end if
```

Algorithm 5 MobCCN_AH: procedure at node p at the expiration of the ageing timeout T_{ageing}

Require: E_p ▷ Set of current neighbours of node p

```
1:  for each e ∈ PIT do
2:      if not PIT.Expired(e) then
3:          i ← PIT.Name(e)
4:          if FIB.Find(i) then
5:              find the node j with the highest utility for content type i in FIB and
6:              CNU tables
7:              if j ∈ E_p  and  U_{j,i} > U_{p,i} * (1 + Hyst/100) then
8:                  forward Interest packet to node j
9:              else
10:                 store Interest packet waiting for the next contact
11:             end if
12:         else
13:             drop Interest packet
14:         end if
15:     else
16:         drop Interest packet
17:         PIT.remove(e)
18:     end if
19: end for
```

supporting various hardware/software platforms, including the OMNeT++ simulator.

In the simulated environment, we investigate a basic scenario composed of 60 nodes randomly located in a square area of 1000 m × 1000 m. We assume that a short-range wireless communication technology is used to opportunistically establish D2D links between nearby users' devices. This leads to a sparse network scenario with potentially short connection times, which is a challenging condition for any opportunistic data dissemination protocol (for details on simulation settings see Table 1). If not otherwise stated we assume that the users move with an average speed of 1.5 m/s (a typical walking speed) according to the Home-cell Community-based Mobility Model (HCMM). HCMM is a popular human mobility model [51] able to reproduce realistic mobility patterns between the individuals of different social communities, as well as the opportunistic contacts of their personal devices. It is important to point out that we do not simulate the exact positions of nodes, but only the communication events between pairs of adjacent nodes using the starting and ending time instants of their opportunistic contacts. In addition, we do not simulate the communications between the static IoT devices and the users' mobile devices, but we consider as initial system state the time when mobile devices have already collected all the available contents from the IoT devices that are present in the area. In other words, we assume that when a requester issues

Table 1
Simulation settings for the basic scenario.

Mobility parameters	
Area	1 km × 1 km
Mobile nodes	60
Mobile communities	3
Travellers	3
Communication range	10 m
Mobility model	HCCM [51]
avg. speed	$\mathcal{U}(1, 1.86)$ m/s
Traffic parameters	
Producers	6
Consumers	12, 24
Content types per producer	4
Chunks per content type per producer	5
Requests per consumers	40
Distribution req. times	exponential ($\lambda = 1/900$)
Simulation time	36 h
req. start time	11 h
req. end time	22 h

Fig. 4. Network overview.

an Interest packet for a specific content, that content is already stored in one of the available mobile devices. The direct utility of these nodes that initially store the content is set up to a randomly high value. Finally, we assume that a single copy of that content exists in the network when a request for that content is generated.[4] If not otherwise stated, in the following simulations mobile nodes are grouped into three separate communities, and each individual belongs to a single community and move within it. However, there is a small subset of nodes, called "travellers", which can sporadically visit external communities and allow the dissemination of contents between communities. Without loss of generality, in our study we assume that each community has a single traveller node. This setup represents, for example, users moving in a specific (social) environment, such as their working place, and sometimes moving in other contexts, as a side effect of "external" social relationships. A schematic representation of the resulting network is shown in Fig. 4.

Regarding the traffic patterns, each content producer manages 4 content types, each composed of 5 different chunks, thus storing 20 contents in total, while each content consumer asks for 40 different contents. Among these 40 requests, half of them targets contents that are produced by nodes belonging to the same community of the requester, while the remaining 20 requests are uniformly distributed among the contents produced by the other two external communities. Content requests are generated after a warm-up period and terminate before the simulation end to reduce the probability that packet losses occur because the simulation ends before the data dissemination process is concluded. Content requests are generated following an exponential distribution of inter-request times with parameter $\lambda = 1/900$, which corresponds to a content request every 15 min, on average.

As anticipated, in Section 6 we compare all the MobCCN retransmission mechanisms with benchmarks under different scenarios and working hypothesis. It is useful to remark that the low speed and the size of the simulated area yields to long inter-contact times, which leads to long data dissemination delays. However, delays will scale down is denser network scenarios and/or higher mobility speeds would be considered.

5.1. Performance metrics

The performance comparison is based on the following network-related performance metrics:
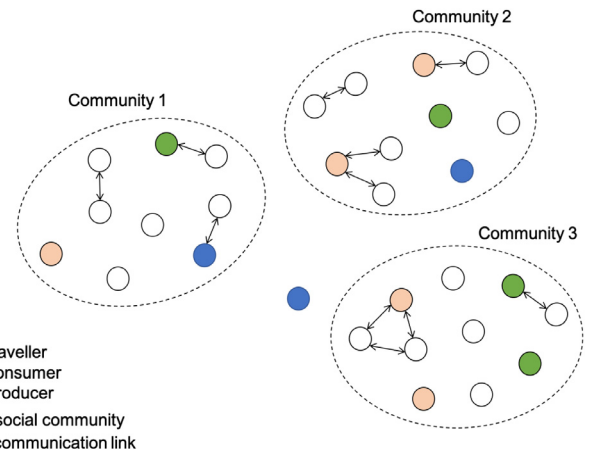
- *Delivery rate*: percentage of Data packets successfully received at the consumers. In scenarios where we assume an infinite bandwidth during contacts among nodes,[5] the only cause of unsuccessful Data delivery is ascribed to the failure of MobCCN in delivering Data packets to consumers before the end of the simulation;
- *End-to-end latency*: time interval between the generation of the Interest packets and the successful reception of the corresponding Data packets (this metric is computed only for those Data packets which have been correctly received by consumers);
- *Hop count*: number of nodes through which Data packets pass in their path from producers to consumers[6];
- *Total traffic*: the total amount of traffic generated by all the nodes. This includes the forwarding traffic composed by Interest and Data packets, and the control traffic due to routing when present.

To better highlight the impact of mobility on system performance, we also measure delivery rate, end-to-end delay and hop count at the "community level". Specifically, for the aforementioned metrics we measure two different components: (i) a "local" one, which refers to the metric result achieved for the requests of contents that are produced by nodes within the same community of the requester, and (ii) an "external" one, which is related to the metric result obtained for the requests of contents held by nodes outside the requester's community.

To quantify the impact of retransmission traffic we introduce the following metrics:

- *Retransmission per Interest*: number of retransmitted Interests per generated requests, which measures the average amount of retransmissions (I_{reTX}) for each Interest generated by a consumer (I_{gen}). It is given by:

$$RPI = \frac{I_{reTX}}{I_{gen}} . \tag{4}$$

- *Retransmission per Data*: number of retransmitted Interests per received content, which measures the average amount of retransmissions needed to successfully receive a Data packet at the consumer. It is given by:

$$RPD = \frac{I_{reTX}}{D_{rcv}} . \tag{5}$$

---

[4] This is equivalent to assume that the IoT devices cancel the content when it is delivered to one of the available users' devices.

[5] This is customary in the opportunistic networking literature.

[6] As Data packets follow the reverse path of their corresponding Interest packets, the hop count also represents the number of hops of Interest packets.

Note that D_{rcv} can be lower than I_{gen} due to a delivery rate lower than one.

Each simulation is replicated 10 times and the performance results are averaged over all the replicas. In the following, average values are shown with 95% confidence interval.

5.2. Benchmarking protocols

We compare MobCCN and its variants with two benchmarks based on the well-known Epidemic routing protocol for opportunistic networks [52]. The first benchmark is the classical Epidemic scheme that works by flooding both Interest and Data packets in the network, i.e., each time two nodes encounter they exchange all their Interest and Data packets. Intuitively, under the simplifying hypothesis of infinite bandwidth and unlimited cache size at the nodes, this protocol provides the shortest and minimum-delay path from a source to a destination, representing an upper bound in terms of delivery rate, and a lower bound in terms of end-to-end latency and hop count. Its major drawback is the high cost in terms of network overhead due to the uncontrolled packet flooding. Including Epidemic in the comparison allows us to understand how much our proposed solutions deviate from the optimal one in terms of delivery rate, end-to-end latency and hop count, and how much they gain in terms of network traffic. In the rest of the paper we refer to it as "*Ideal Epidemic*".

The second benchmark is a *copy-limited* version of Epidemic, which works as follows. Whenever an intermediate node receives an Interest packet, it checks if the Interest packet has been already forwarded. If this is the case, it simply drops the packet, otherwise it forwards it with a probability r (in our experiment we set r to 0.5). Data packets are then routed back to those nodes who requested them exploiting the reverse-paths, as happens for MobCCN. As apparent, this version of Epidemic protocol drastically limits the number of Interest and Data packets that are flooded in the network at the cost of a reduced delivery rate and an increased end-to-end latency and hop count. Including this protocol in the comparison allows to investigate which are pros and cons of the utility-based protocols compared to the flooding-based ones. In the rest of the paper we refer to it as "*Limited Epidemic*".

6. Performance analysis

In this section we compare the performance of the different retransmission mechanisms described in Section 4 against the two benchmark protocols described in Section 5.2, namely Ideal Epidemic and Limited Epidemic.

Before that, we briefly summarize the sensitivity analysis we conducted to fine-tune the protocol parameters. The interest readers may refer to [53] for the complete analysis. In the retransmission mechanisms based on the number of received requests, namely MobCCN_R1 and MobCCN_R2, the parameter that triggers the retransmission of duplicate Interest packets is the retransmission threshold $ReTX_{threshold}$. Our results show that different $ReTX_{threshold}$ values have a small impact on the system performance. We have also noted that there is a small increase of the delivery rate when each duplicate reception of an Interest packet triggers a retransmission, hereafter referred to "0" setting, while the other performance metrics remain almost unchanged. As far as MobCCN_A is concerned, we observed significant difference for the delivery rate (up to 10 percentage points) when varying the parameter that regulates the periodic retransmission of pending Interest packets. It is clear that a sub-optimal setting of the ageing parameter results into a worse protocol performance.
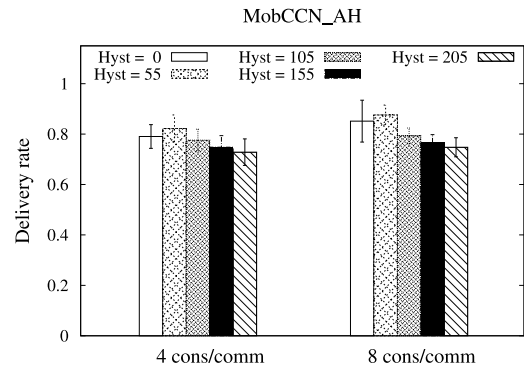

Fig. 5. Sensitivity of MobCCN_AH on *Hyst*: delivery rate in 4 consumers/community and 8 consumers/community configurations ($T_{ageing} = 40\,000$).
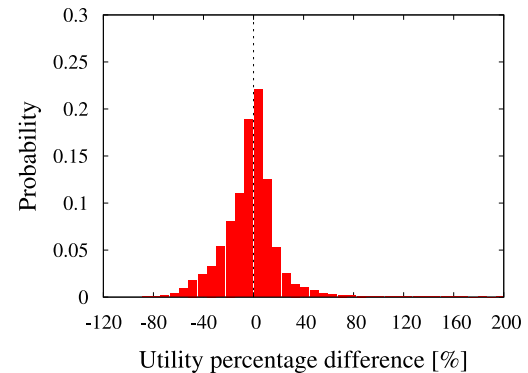

**Fig. 6.** Distribution of the utility percentage differences among nodes.

Table 2
MobCCNs settings.

MobCCN version	Parameter
MobCCN_R1	$ReTX_{threshold} = 0$
MobCCN_R2	$ReTX_{threshold} = 0$
MobCCN_A	$T_{ageing} = 40\,000$
MobCCN_AH	$T_{ageing} = 40\,000$, Hyst = 55%

In our scenario the best configuration is $T_{ageing} = 40\,000$. Concerning the parameters specific to MobCCN_AH, i.e., the hysteresis percentage *Hyst*, we observed that the delivery rate initially increases, reaches a peak value and, then, start decreasing (see Fig. 5). The maximum delivery rate is achieved with a hysteresis percentage equal to 55%. This means that the strategy of delaying the retransmission of the pending Interests is beneficial, but up to a certain extent. The delivery rate gain is up to 5% in such cases. On the contrary, postponing too much the forwarding in order to find a node whose utility is significantly higher is counterproductive. For instance, waiting for a node whose utility is two times higher leads to lower delivery rates, even lower than what is obtained with MobCCN_A (first column). This can be explained by observing the distribution of utility percentage differences between encountered nodes in Fig. 6, where it is shown that the majority of nodes have utility at most one time higher, while most of the distribution lies within the range [−50%, 30%]. Thus, setting the hysteresis threshold to a high value may have the negative effect of introducing an excessive delay before triggering a retransmission of an Interest packet.

In the rest of the section we compare the performance of all the MobCCN variants against the two benchmarks taking into account the above results. Specifically, each MobCCN variant is
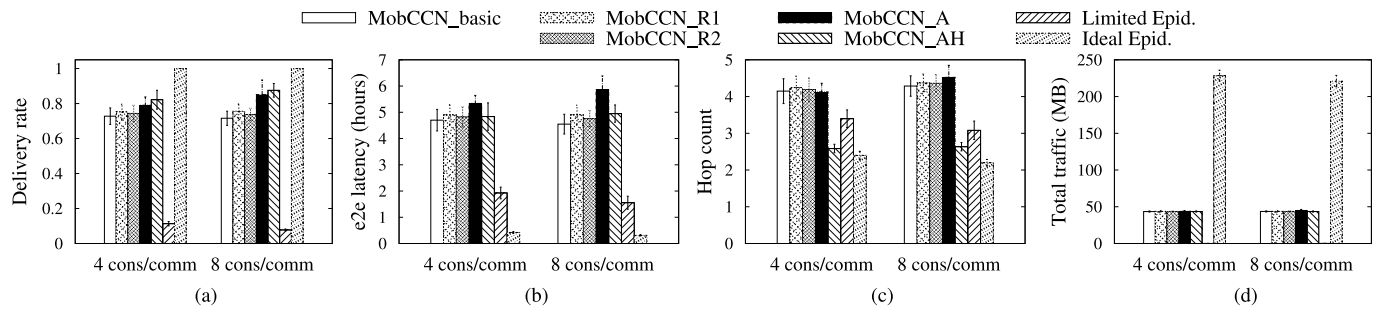
Fig. 7. Network-level performance: delivery rate (7a), end-to-end latency (7b), number of hops (7c), and total traffic (7d) in 4 consumers/community and 8 consumers/community configurations.

simulated using the best configuration for its protocol parameter(s) that is obtained when the delivery rate is maximized. Table 2 summarizes the setting used.

We split the evaluation into five parts. In Section 6.1 we focus on the basic scenario, whose parameter settings are depicted in Table 1. Here, we first focus on the aggregate behaviours of the protocols by analysing the metrics at the network-level. Then, we increase the detail of the analysis by investigating what happens at the community level. To investigate the protocols performance under different traffic patterns, in Section 6.2 we change the number of consumers requesting content (i.e., increase of traffic load), while in Section 6.3 we analyse a scenario where there is an overlapping between content requests of different consumers (i.e., increase of content reuse). In Section 6.4 we show more detailed results on the amount and distribution of retransmission traffic generated by the MobCCN retransmission mechanisms.

It is important to point out that the results in Sections 6.1–6.4 are obtained under the simplified hypothesis of infinite communication bandwidth, as we wanted to measure the number of failed content requests due to protocol inefficiency and mobility patterns without the effect of bandwidth bottlenecks. In Section 6.5 we relax such assumption and we also investigate the system performance under limited network bandwidth.

6.1. Basic scenario

6.1.1. Network-level performance

Figs. 7 illustrates the obtained results for all the performance metrics at the network level in the basic scenario. Looking at the delivery rate (see Fig. 7a), Ideal Epidemic achieves the best performance with a delivery rate equal to 100%, as expected. By contrast, Limited Epidemic results to be the worst scheme with a delivery rate below 10%. The main reason for such a low performance is the probabilistic strategy used for forwarding Interest packets, which can lead to useless transmissions of the same Interest packet. Conversely, in the same conditions, MobCCN significantly outperforms Limited Epidemic with delivery rates higher than 70%, thus confirming that the use of the utilities provides a more efficient way of searching for requested content. As far as the MobCCN variants, we observe the following delivery rate for the "4 cons/comm" configuration: 75% for MobCCN_R1, 74% for MobCCN_R2, 79% for MobCCN_A, and 82% for MobCCN_AH. The figure clearly highlights that transmission reliability greatly improves with retransmissions, resulting in an increase of delivery rates. However, such advantages are marginal if the retransmission strategy relies only on the number of the received requests. This is independent of the type of paths followed, either be they optimal as in MobCCN_R1 or sub-optimal as in MobCCN_R2. Indeed in such cases, since the retransmissions follow the packet arrival rate, they occur close together in time (within two hours from each other). As consequence,

the utility values of the possible forwarders that can be used to find a requested content change little. This means that, in case of MobCCN_R1, the forwarder with the highest utility towards a specific content is likely to remain the same node of the first transmission. As far as MobCCN_R2, a different node is picked at least at the first hop, as the second highest utility is used. But, in general, the paths used by retransmitted packets are mostly overlaid with those used in the first transmission. On the contrary, advantages are more significant if the retransmission is executed on a periodic basis, as the delivery gains are in the range of 10%–15%. This is due to the fact that, in this second case, as retransmissions occur after longer times, the network reorganizes itself and the nodes identify new best forwarders. Therefore, retransmissions explore new paths, separated from those used in the first transmission. It is worth pointing out that, among the two periodic-based protocols, MobCCN_AH reaches better performance. The hysteresis mechanism indeed ensures that the nodes look for the nodes to which to forward/retransmit their requests more carefully, forwarding them only when there is a significant advantage in terms of utility. Another observation is related to the increase of the delivery rate when the number of consumers grows from four to eight. The more the consumers, and the more the number of content requests to satisfy, but the higher is also the probability that requests are partially overlapped. Therefore, one retransmission may satisfy more requests at the same time due to the aggregation mechanism of Interest packets used in the PIT. Again, MobCCN_AH outperforms the other MobCCN versions.

Average e2e latencies are depicted in Fig. 7b. Also in this case, Ideal Epidemic outperforms all the other protocols experiencing very low delays. This is expected as, in order to find the requested contents, it spreads the Interests on all the nodes encountered. Low delays are also experienced by Limited Epidemic, but this depends mainly on two factors: (i) this average is computed from a low number of successfully received Data packets, and (ii) those Data packets mainly refer to contents belonging to the local consumers' communities and thus experience shorter latencies. This aspect will be further examined in the next section. As far as MobCCN and its variants, e2e latencies are in the range of [4.7–5.3] h in the "4 cons/comm" configuration and in the range of [4.55–5.86] h in the "8 cons/comm" configuration. Concerning the two periodic-based protocols, we observe that MobCCN_A has the largest average latency to retrieve the contents, while MobCCN_AH performs better with a latency only a bit higher than MobCCN_basic. Also in this case, we observe that the hysteresis mechanism helps in improving the performance lowering the end-to-end latencies.

In Fig. 7c the average hop count is reported. As far as this metric, the graph highlights an important MobCCN_AH feature. On the one hand, it outperforms all the other MobCCN versions by cutting the paths length by half. On the other hand, its performance is almost equivalent to that of Ideal Epidemic, with about 2.5 hops on average. This is an important result that highlights
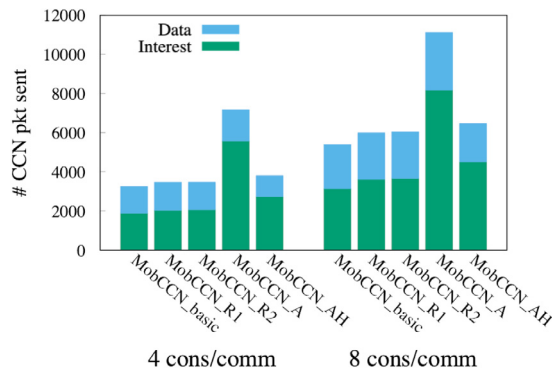
Fig. 8. Breakdown of the forwarding component for the MobCCN protocols.

Fig. 9. Community-level performance: delivery rate for the local community (9a), and the external communities (9b).

6.1.2. Community-level performance

In this section we focus on the protocols performance at the community level, that is, as described in Section 5.1, we split the performance indexes in two components according to the membership of the requested contents, be they local or external, and we analyse them separately. We recall here that every consumer makes 40 content requests such that 20 are related to contents inside its own community and the remaining 20 are uniformly distributed among the other two external communities.

Fig. 9 shows the two separated delivery rate components: the local one in Fig. 9a, and the external one in Fig. 9b. As highlighted by the figure on the left, Limited Epidemic has the lowest performance, with delivery rates of 22% and 13% in the "4 cons/comm" and "8 cons/comm" configuration, respectively. This is due to the fact that in Limited Epidemic nodes are all equivalent and the search of the forwarder to which to send the request is random. On the contrary, the delivery rate performance rises significantly when MobCCN and its variants are used because here nodes have different utility weights. MobCCN alone (i.e., MobCCN_basic) reaches values above 95% in both configurations, confirming that the utility mechanism is a successful strategy. Also, when retransmission mechanisms are running, the protocol reliability further increases, with MobCCN_AH that reaches a 100% delivery rate retrieving all the contents belonging to the local community of consumers, equal to Ideal Epidemic. Similar considerations can be drawn for the external component of the delivery rate as depicted by Fig. 9b, although the numbers involved are lower. The unique value that remains unchanged is the one related to Ideal Epidemic, which is able to deliver successfully all the contents requested by nodes in external communities. By contrast, Limited Epidemic's performance drops further with only one satisfied request, on average. Concerning the other protocols, we observe that at least 50% of the external content requests are satisfied with MobCCN and its variants. In addition, this figure clearly highlights that the retransmission mechanisms introduced, although useful for performance, affect differently the delivery rate. Specifically, compared to MobCCN_basic, MobCCN_R1 and MobCCN_R2 increase the delivery rate by a few percentage points only. On the contrary, the delivery gain obtained with MobCCN_A and MobCCN_AH is higher, in the order of [12–30] percentage points. Further observations can be then derived for each retransmission scheme. First, MobCCN_R1 performs slightly better than MobCCN_R2, that is, retransmitting over the same best path guarantees higher performance with respect to exploiting multi-paths that pass through non-optimal forwarders, as happen in MobCCN_R2. Second, MobCCN_AH has the best performance with delivery rates equal to 64% and 74% in the "4 cons/comm" and "8 cons/comm" configuration, respectively. This result indicates that combining a periodic retransmission based on a large retransmission timeout and a hysteresis mechanism significantly

the ability of MobCCN_AH to achieve excellent delivery rates with low resource consumption, in contrast to Ideal Epidemic that, on the contrary, needs to replicate packets over the network, hence wasting useful resources.

Fig. 7d shows the total traffic for all the protocols. We recall here that the total traffic metric includes all the traffic generated by all the nodes, thus for the MobCCN versions this includes not only the Interest and Data packets as in Ideal Epidemic or Limited Epidemic, but also the control traffic due to the routing protocol. As shown by the figure, Ideal Epidemic has the largest total traffic which can be as high as 230 MB, followed by all the MobCCN versions with less than 50 MB, and by the Limited Epidemic, whose traffic is almost negligible. The MobCCN versions therefore perform very well, significantly reducing network traffic compared to Ideal Epidemic, with a 78% traffic cut. Obviously, they produce more traffic over the network than Limited Epidemic, mainly due to the routing information that needs to be sent in order to populate FIB tables, but it is a fair compromise that permits to obtain good delivery rates (see Fig. 7a).

Finally, to investigate more in depth the impact of the retransmission mechanisms on the protocol overheads, Fig. 8 shows the number of Interest packets and Data packets that are produced by each MobCCN variant. We note that MobCCN_basic sends the lowest number of Interests, followed by MobCCN_R1 and MobCCN_R2 with similar aggregate number of forwarding traffic, and then by MobCCN_AH with a 18% increase compared to MobCCN_basic. On the contrary, MobCCN_A has the worst performance by doubling the number of Interests with respect to the other MobCCN versions. On the contrary, the behaviour of Data packets is different. In this case, MobCCN_A still shows the worst performance but the difference with the other protocols is less remarkable. MobCCN_AH achieves the better performance reducing the number of Data sent by 17% with respect to MobCCN. Reducing the number of Data packets as done by MobCCN_AH is very useful, and such advantage grows as the data transmitted increases. This essentially happens for two reasons. First, since the size of a Data packet is on average much greater than the one of Interest packets, as it consists of a few bytes mainly related to the identification string of the content to search, the Data component plays a key role compared to the Interest one. Thus, a reduction of the number of transmitted Data packets corresponds to a significant decrease in the amount of forwarding traffic sent compared to other protocols. Second, as soon as the data traffic intensifies, the forwarding component becomes predominant over the routing one. There is therefore a consequent reduction of the overhead necessary to guarantee the correct population of the FIB tables, and more generally of the total traffic sent.
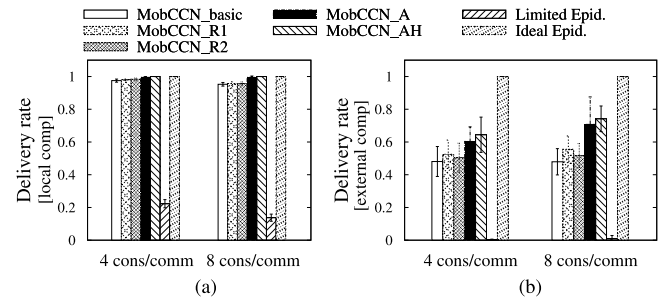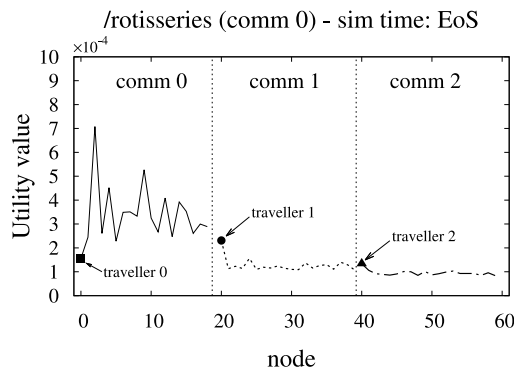
Fig. 10. Example of the utility values towards a content belonging to community 0 (i.e., "rotisseries") at the end of the simulation. Nodes of community 0, which are shown on the left side of the figure, experience high non-uniform utility values as they meet directly the content producer but with different mobility patterns. The utility value of the traveller of community 0 is, on the contrary, the lowest because it also visits communities 1 and 2, thus reducing the probability to meet the content producer. Similar utilities are experienced also by travellers of communities 1 and 2, which commute by definition among all the three communities. Nodes outside community 0 experience very low utilities because they never meet directly the content provider. Note that all the three travellers have utility values of about 50% higher than the utilities of nodes within community 1 and 2.
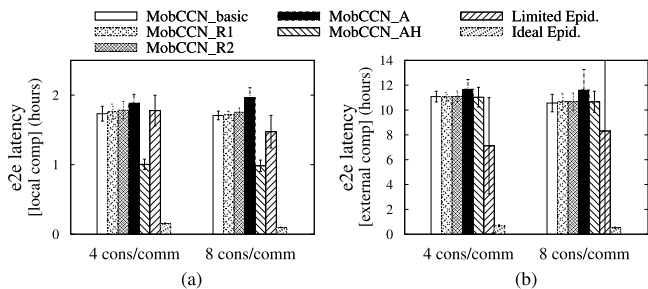
Fig. 11. Community-level performance: end-to-end delay for the local community (11a), and the external communities (11b).

increases the transmissions reliability, especially for those contents belonging to external communities. In this scenario, indeed, it is essential to detect those nodes that can contribute the most to the content dissemination process, that is, the *travellers*, which are the only mobile devices able to retrieve external contents as they commute among communities. As shown in Fig. 10, the travellers have utility values towards contents of external communities much higher than the corresponding utilities of nodes inside the local community (of about 50%). By measuring the utility difference among the nodes in contact, as performed by the hysteresis condition, MobCCN_AH is able to establish whether or not the encountered node is a traveller. When the utility difference is minimal, the node in contact is a common node. On the contrary, when this difference becomes significant and higher than the hysteresis value, the node in contact is a traveller. The latter is then chosen by the forwarding process and exploited to retrieve the desired external contents.

Fig. 11 shows the two end-to-end latency components. For what concerns the local component, Ideal Epidemic introduces the smallest average time to satisfy the data requests, followed by MobCCN_AH with one hour on average, and then by all the other protocols. When the number of Interest requests increases, the network load becomes higher, however the latency delays show no significant change. Note also that MobCCN_AH halves the latency with respect to MobCCN_A, highlighting once again its advantages. For what concerns the external components, Ideal
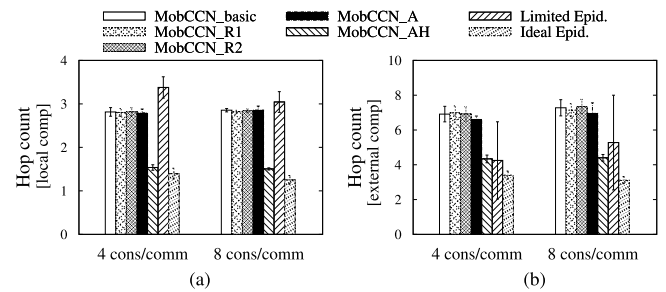
Fig. 12. Community-level performance: hop count for the local community (12a), and the external communities (12b).

Epidemic, by replicating Interests on every encountered node, experiences very low latencies also to retrieve external contents. On the contrary, all the MobCCN variants have similar performance but experiencing higher delays. Also in this case, MobCCN_AH slightly lowers the delays with respect to MobCCN_A. Limited Epidemic latency is statistically insignificant as it refers to a single sample (the external delivery rate is near 0, as shown by Fig. 9b).

Finally, in Fig. 12 the two components of the hop count metric are depicted. Both components show a similar behaviour, both when retrieving local contents and external contents. Obviously, the paths used to retrieve local contents are shorter than those for retrieving external contents. Overall, the group of protocols with the shortest paths length is composed by MobCCN_AH and Ideal Epidemic, followed by the group composed of the remaining MobCCN protocols (i.e., MobCCN_basic, MobCCN_R1, MobCCN_R2 and MobCCN_A) that achieve similar performance. Finally, Limited Epidemic has the worst performance using the longest paths (note that also here the foreign component is insignificant as it refers to a single sample). This figure highlights the ability of MobCCN_AH to halve the paths length compared to all the other MobCCN solutions, getting closer to the minimum paths detected by Ideal Epidemic, all this by lowering significantly the overall network traffic. Therefore, postponing the transmission in order to find the optimal forwarder avoids the unpleasant ping-pong situations among nodes, and allows to exactly identify those nodes useful for a successful Data recovery, such as for example the travellers for retrieving external contents. As a result, paths are shorter and resources consumption is reduced.

Also this community-level analysis confirms that a smart retransmission mechanism helps in improving the MobCCN performance. Specifically, we observe that MobCCN_AH, with its combination of periodic-based retransmission and hysteresis mechanism, is able to achieve good performance results getting very close to the Ideal Epidemic performance for satisfying requests of local contents. Moreover, it achieves the best performance with respect to the remaining MobCCN variants when retrieving contents of external communities. This is also a good result given that the contacts between the communities, consisting of the travellers movements only, are very limited since there is only 1 traveller per community.

6.2. Impact of network load

Fig. 13 illustrates the performance of the considered protocols at the network level when varying the network load. Specifically, we increase the number of consumers generating Interests from 4 up to 16 consumers per community.

Looking at Fig. 13a, we observe that in general the periodic retransmission approach continues to perform better than the retransmission approach based on the number of received requests also when the traffic load increases, with an average gain
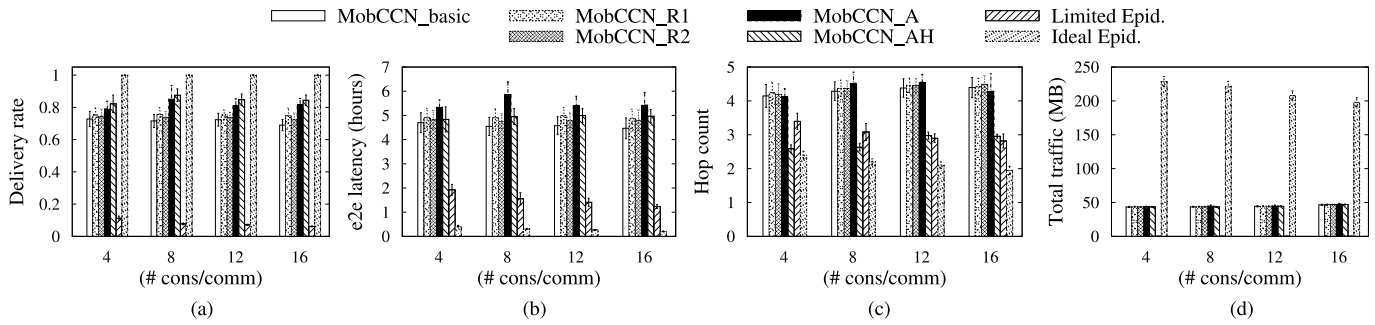
Fig. 13. Network-level performance in the scenario with an increased network load: delivery rate (13a), end-to-end latency (13b), number of hops (13c), and total traffic (13d) for different consumers/community configurations.
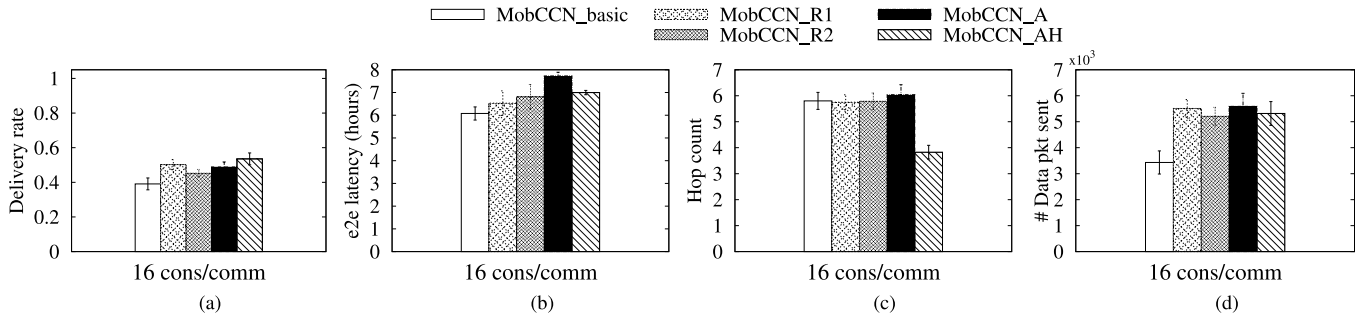
Fig. 14. Network-level performance in the scenario with a different request pattern: delivery rate (14a), end-to-end latency (14b), number of hops (14c), and total traffic (14d) in the 16 consumers/community configuration.

in the range of [10–15]%. Again, MobCCN_AH achieves higher performance than the MobCCN_A counterpart, hence confirming its efficiency in the different network conditions we tested. Fig. 13b shows that MobCCN_A experiences higher end-to-end latencies with respect to the rest of MobCCN variants, which are, on the contrary, almost aligned around 5 h. MobCCN_basic latencies are lower but this is essentially due to a lower number of received contents. As far as the hop count metric (see Fig. 13c), we observe that MobCCN_AH significantly reduces the path lengths with respect to the other MobCCN retransmission schemes, with a reduction of around 33%. In Fig. 13d we observe that the total traffic for all the MobCCN versions is basically the same and stabilized around less than 50MB. When increasing the number of consumers, we also note a slight rise of the traffic, which is basically due to the higher number of Interests generated. On the contrary, Ideal Epidemic exhibits a decreasing trend. The same behaviour is apparent also for the other performance metrics. Since there are overlapping requests, they spread even faster with the growing number of consumers. This is a consequence of how Interests are replicated at nodes. Indeed, in Ideal Epidemic, a node floods Interests to all the nodes it is in contact with. Therefore, the greater is the number of the requests for the same content, the higher is the speed with which that Interest is spread in the network. As a result, in Ideal Epidemic contents are retrieved faster, cross fewer hops, and the network traffic is slightly reduced.

As far as the performance at the community level, we observe a similar trend under these scenarios, and thus we omit the corresponding plots here.

6.3. Impact of content request pattern

In this section we report the performance results of the scenario in which we vary both the pattern of content requests and the traffic load. Specifically, we analyse a scenario with 16 consumers per community where each consumer asks for the same set of contents, that is, content requests are completely overlapped. The requested set consists of 80 different contents

divided, as before, between their own community (40) and the two external ones (40).

Looking at the delivery rates shown in Fig. 14a, we note that they are significantly lower than those in the previously considered scenarios being in the range [0.4, 0.55]. However, MobCCN_AH continues to achieve the highest delivery performance with a delivery gain of 15% with respect to MobCCN_basic. As far as the e2e latencies (see Fig. 14b), they range in the [6 h–8 h] interval. MobCCN_A experiences again the highest delays for retrieving content, followed by MobCCN_AH and MobCCN_R2 with similar values, then MobCCN_R1 and finally MobCCN_basic. However, MobCCN_basic experiences also the lowest number of received contents. Fig. 14c, which shows the average path length, highlights once again the capability of MobCCN_AH to reduce it by more than 30% with respect to the other protocols. Finally, as a measure of the network traffic, in Fig. 14d we report the forwarding traffic component related to the Data packets sent, being the one that influences the most the amount of forwarding traffic sent. Excluding MobCCN_basic, whose lower traffic is mainly due to the lower delivery rate, also in this case MobCCN_AH achieves good performance reducing the number of Data packets sent with respect to the other protocols.

This scenario, which is in general advantageous for protocols that make use of in-networking caching (even temporarily), confirms the ability of MobCCN_AH to find the desired contents more precisely and with higher reliability than the others.

6.4. Retransmission performance

In this section we analyse more in detail the properties of the retransmission procedures implemented by MobCCN_A and MoBCCN_AH, being the two MobCCN variants that maximize the delivery rate, thus guaranteeing the highest reliability performance. To this end, we introduce to new metrics to assess the efficiency of the retransmissions process: namely retransmission per Interest (RPI) and retransmission per Data (RPD). We focus on a scenario with a varying number of consumers per social community, in the range from 4 to 16 consumers.
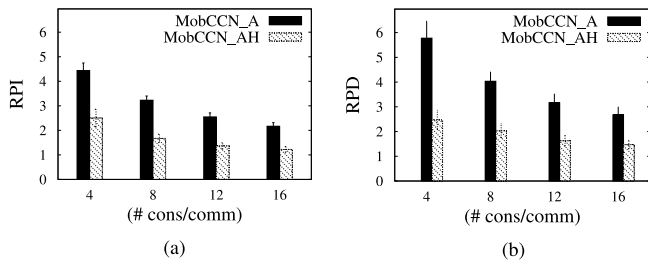
Fig. 15. Retransmission performance: retransmission per Interest (RPI) (15(a)), and retransmission per Data (RPD) (15(b)) for different consumers/community configurations.
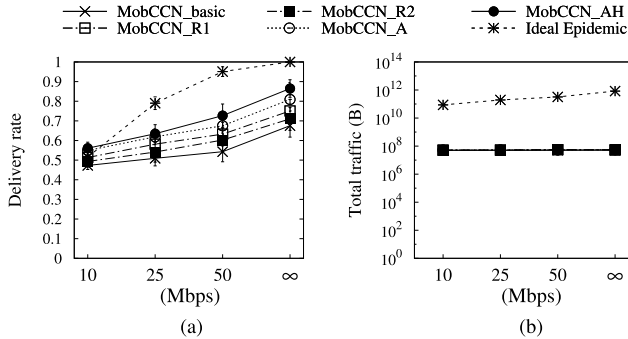

Fig. 16. Delivery rate (16a) and total traffic (16b) when varying the available bandwidth.

Fig. 15(a) shows the RPI value for the evaluated scenarios, and we observe that MobCCN_AH significantly outperforms MobCCN_A in all the four configurations we considered. Specifically, the number of retransmissions for each Interest generated by the consumers is very limited, with at most 2 retransmissions in the case of 4 consumers per community. On the contrary, MobCCN_A needs around 5 retransmissions per Interest in the same scenario. Another observation is that both protocols reduce the number of retransmissions as the number of consumers increases. This is another indication of the multiplexing effect achieved by a content centric data-dissemination protocol when the number of interested nodes increases. As far as RPD (see Fig. 15(b)), we can draw similar considerations, and MobCCN_AH outperforms MobCCN_A with a significant drop of the number of retransmissions per Data received, which is estimated between 40% and 58%. For instance, in the worst case MobCCN_AH generates less than 3 retransmissions per each successfully received Data packet, while MobCCN_A requires more that 6 retransmissions for the same outcome. Also for this metric we observe that it is inversely proportional to the number of consumers.

From the above discussion, we can conclude that another important advantage of MobCCN_AH is that not only it achieves the highest performance compared to the other retransmission mechanisms that we have explored, but it also reaches such performance by spending less in terms of retransmissions, and consequently limiting further the network resource consumption.

6.5. Impact of bandwidth

The results shown so far are obtained by assuming the bandwidth of D2D links is infinite, namely all the chunks stored in the node can be transmitted instantaneously and without losses. In this section, we complete our performance analysis by removing such simplifying assumption. More precisely, we assume that the bandwidth of the wireless channel is a fixed and constant value, and that a perfect channel access scheme is used that allows equally sharing the channel bandwidth among the nodes that are
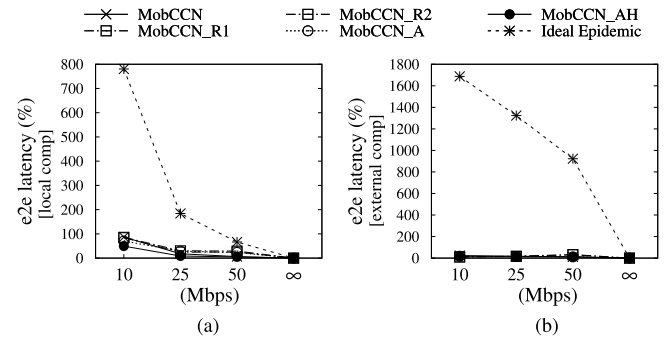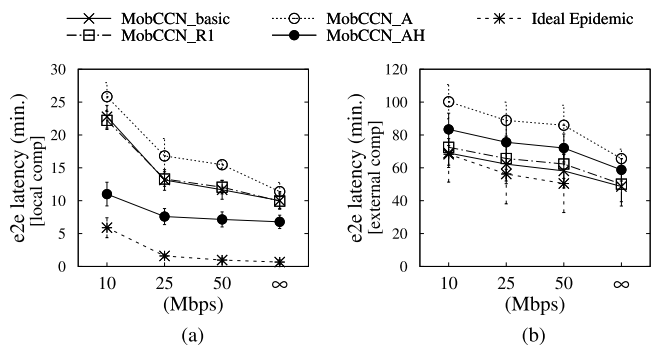

Fig. 17. Breakdown of the end-to-end latency when varying the available bandwidth: percentage increase of the end-to-end latency with respect to the infinite bandwidth case for the local component (17a) and external component (17b).

in contact. In the following tests we set the size of each chunk equal to 2 MB, and we vary the channel bandwidth from 10 Mbps to 50 Mbps.

Looking at the delivery rates (see Fig. 16a), we observe that reducing the channel bandwidth results in an overall decrease of the delivery rate for all the protocols, as expected. Two main reasons can explain this result. First of all, the more the neighbours, the lower the available per-node communication bandwidth. Second, a contact may not be sufficient for transmitting all the available contents stored on the node, and a transmitter should wait for the next contact with the same node to complete the transmission. The behaviour of the delivery rate is similar for all the MobCCN variants, and the curves tend to converge to values in the range [0.45–0.55] with 10 Mbps bandwidth. MobCCN_AH still achieves the highest (but reduced) delivery rate, about 10% higher than the one of MobCCN_basic in the worst case with 10 Mbps bandwidth. Delivery rate degrades also for Ideal Epidemic, down to 0.49 in the 10 Mbps case, thus losing its advantage over the MobCCN variants. This behaviour is justified by the huge amount of sent traffic (around 775 GB as highlighted by Fig. 16b) that excessively consumes the available bandwidth. On the contrary, this effect is less marked with the MobCCN variants since the volume of sent traffic is significantly lower, being stable around 50 MB. As far as the e2e delays, instead of reporting their absolute values, we calculate the percentage increase with respect to the reference value of the infinite bandwidth, splitting them into the local and the external components. As shown in Fig. 17, the bandwidth constraint has a less noticeable impact on the delay performance for the MobCCN variants, and delay increases range from 25% to 80%. On the contrary, the bandwidth limitation has a more negative effect for Ideal Epidemic, as e2e delay increases almost 1800% with respect to the reference value in the 10 Mbps case.

Interestingly, the above results have shown that a bandwidth limitation have a small impact on the e2e delay. This implicitly suggests that the main reasons for the very high delays depend on mobility patterns, namely the very low frequency with which encounters happen and low network density. To confirm this intuition, we also investigate a more dynamic scenario where the inter-contact times between the nodes are considerably reduced. For this purpose, we have manipulated the mobility traces used in the previous tests by setting the inter-contact times to 10% of the initial ones. Then, in addition to modifying the content request patterns to be in line with the mobility trace obtained, we have performed a new sensitivity evaluation of the protocol parameters since the nodes meet more frequently and set them accordingly, i.e., $ReTX_{threshold} = 0$, $T_{ageing} = 1800$, and $Hyst = 35\%$. As before, we evaluate the system performance under bandwidth constraint assumption. As far as the delivery rates, the results

Fig. 18. Breakdown of the end-to-end latency in the scenario with reduced inter-contact times: local component (18a) and external component (18b) when varying the available bandwidth.
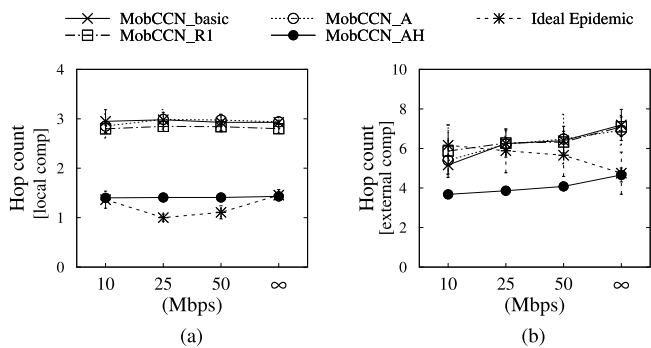
Fig. 19. Breakdown of the hop count in the scenario with reduced inter-contact times: local component (19a) and external component (19b) when varying the available bandwidth.

are aligned with the ones shown in Fig. 16a and are omitted. Comparing the latencies (see Fig. 18), we observe that: (i) reducing the available bandwidth results in an increase of the e2e delays, but (ii) the latter are significantly reduced both for local requests and for the external ones with respect to the previous set of experiments. The more frequent are the node encounters, the more densely connected is the network and the higher is the latency decrease. Again, Ideal Epidemic experiences the lowest values in both cases, while MobCCN_AH performs better among the MobCCN family in the local component. This does not hold for the external component where, on the contrary, MobCCN_AH achieves an intermediate performance while MobCCN_basic results are the best ones. However, such result is statistically biased as the values over which the delays are averaged are not comparable in the two cases as MobCCN_basic satisfies only a very low number of external requests. Finally, the hop count metric is shown in Fig. 19. As far as the local component (left graph), we observe that path lengths in MobCCN_AH are comparable with those in Ideal Epidemic with values of 1.5 on average. The path lengths of the remaining MobCCN variants are longer with average values of 3. As far as the external component (see Fig. 19b), once again it emerges the MobCC_AH ability to choose shorter paths, not only with respect to the other MobCCN variants, but also with respect to the Ideal Epidemic. Therefore, postponing the Interest retransmission is advantageous as it allows the nodes to find a more reliable forwarder, namely the traveller.

7. Conclusions

In this paper, we have investigated the efficiency of different retransmission approaches to improve the reliability of ICN-based networking protocols for IoT environments. To this end, we have extended the reliability of the data retrieval process of MobCCN, an ICN-based data-centric protocol that we proposed in [12,13]

for a mixed IoT environment composed by static and mobile nodes that meet opportunistically and implement ICN functions. This study focused on the analysis of two different classes of retransmission mechanisms. The first class relies on the number of received Interests for the same content to activate the retransmission, while the second relies on the periodic retransmissions of pending Interests. Within each class we then have explored two variants. In the first class, we have considered a single-path approach and a multi-path approach. In the second class, we have studied a retransmission solution based on timeout expiration and one that combines it with a hysteresis mechanism.

Our extensive simulations show that a suitable retransmission strategy is able to improve the reliability of MobCCN while further reducing the network resource consumption. Specifically, the retransmission schemes that employ periodic retransmissions of pending Interests outperform the original MobCCN as well as the retransmission schemes based only the number of received Interests. In particular, the solution that combines both periodic retransmissions and a hysteresis mechanism so as to delay a retransmission if the available next-hop forwarder provides only a marginal utility gain, called MobCCN_AH, achieves the best performance. Indeed, the combination of periodic retransmissions with a careful selection of the best forwarder on the basis of the utility difference among nodes, guarantees that transmissions/retransmissions take place only if needed, that is, if there exists an effective advantage in terms of utility to find the desired content.

In conclusion, the major advantages of MobCCN_AH are: (i) average delivery rates in the order of [85–95]% in case of infinite bandwidth, with a significant improvement of the number of contents retrieved from external communities (up to 60%), and average delivery rates of 70% with 50 Mbps bandwidth; (ii) a 50% reduction with respect to other MobCCN variants of the length of the network path that are traversed by Data packets, closely approaching the average path lengths that are observed with an Ideal Epidemic protocol; (iii) limited number of retransmissions issued per Interest generated and per Data received. The price to pay is the overhead due to the proactive routing traffic required to maintain updated FIBs, which, however, remains reasonably sized and never exceeds 50 MB. Furthermore, this traffic component becomes marginal as soon as the data traffic is more intensive. Another downside is a lower delivery rate performance in cases of channels with bandwidth lower than 25 Mbps. In order to guarantee higher delivery rates even when the channel is congested, and hence making MobCCN_AH more flexible and versatile to the various wireless transmission protocols, additional strategies are required. A possible solution is to study adaptive retransmission mechanisms that better follow the evolution of the network by changing for example the timeout parameter. As future works, we will investigate in-networking caching techniques combined with the MobCCN_AH routing process to further reduce latencies in retrieving contents.

CRediT authorship contribution statement

Eleonora Borgia: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Raffaele Bruno:** Conceptualization, Formal analysis, Writing – review & editing. **Andrea Passarella:** Conceptualization, Formal analysis, Writing – review & editing.

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.

Funding

This work is partly funded by the European Commission under the H2020 INFRADEV-2019-3 SLICES-DS (951850), the H2020 INFRAIA-02-2020 SLICES-SC (101008468), and the H2020 ICT-2020-1 MARVEL (957337) projects.

References

[1] E. Borgia, The Internet of Things vision: Key features, applications and open issues, Elsevier Comput. Commun. 54 (2014) 1–31.

[2] M. Di Francesco, S.K. Das, G. Anastasi, Data collection in wireless sensor networks with mobile elements: A survey, ACM Trans. Sensor Netw. 8 (1) (2011).

[3] E. Borgia, R. Bruno, M. Conti, D. Mascitti, A. Passarella, Mobile edge clouds for information-centric IoT services, in: 2016 IEEE Symposium on Computers and Communication (ISCC), 2016, pp. 422–428, http://dx.doi.org/10.1109/ISCC.2016.7543776.

[4] K.L.-M. Ang, J.K.P. Seng, A.M. Zungeru, Optimizing energy consumption for big data collection in large-scale wireless sensor networks with mobile collectors, IEEE Syst. J. 12 (1) (2018) 616–626.

[5] X. Lin, J.G. Andrews, A. Ghosh, R. Ratasuk, An overview of 3GPP device-to-device proximity services, IEEE Commun. Mag. 52 (4) (2014) 40–48, http://dx.doi.org/10.1109/MCOM.2014.6807945.

[6] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, M.A. Javed, A survey of device-to-device communications: Research issues and challenges, IEEE Commun. Surv. Tutor. 20 (3) (2018) 2133–2168, http://dx.doi.org/10.1109/COMST.2018.2828120.

[7] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, A survey of information-centric networking research, IEEE Commun. Surv. Tutor. 16 (2) (2014).

[8] M. Amadeo, C. Campolo, A. Iera, A. Molinaro, Named data networking for IoT: An architectural perspective, in: 2014 European Conference on Networks and Communications (EuCNC), 2014, pp. 1–5.

[9] E. Baccelli, C. Mehlis, O. Hahm, T.C. Schmidt, M. Wählisch, Information centric networking in the IoT: Experiments with NDN in the wild, in: Proc. of the 1st International Conference on Information-Centric Networking, in: ICN '14, ACM, New York, NY, USA, 2014, pp. 77–86.

[10] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, R.L. Braynard, Networking named content, in: Proc. of the 5th International Conference on Emerging Networking Experiments and Technologies, in: CoNEXT '09, ACM, New York, NY, USA, 2009, pp. 1–12.

[11] 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.

[12] E. Borgia, R. Bruno, A. Passarella, MobCCN: A CCN-compliant protocol for data collection with opportunistic contacts in IoT environments, in: Proc. of the 11th ACM Workshop on Challenged Networks, in: CHANTS '16, ACM, New York, NY, USA, 2016, pp. 63–68.

[13] E. Borgia, R. Bruno, A. Passarella, Making opportunistic networks in IoT environments CCN-ready: A performance evaluation of the MobCCN protocol, Elsevier Comput. Commun. 123 (2018) 81–96.

[14] A.J. Abu, B. Bensaou, J.M. Wang, Interest packets retransmission in lossy CCN networks and its impact on network performance, in: Proc. of the 1st ACM Conference on Information-Centric Networking, 2014, pp. 167–176.

[15] 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.

[16] S. Arshad, M.A. Azam, M.H. Rehmani, J. Loo, Recent advances in information-centric networking-based internet of things (ICN-IoT), IEEE Internet Things J. 6 (2) (2019) 2128–2158.

[17] 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, Elsevier Comput. Commun. 142–143 (2019) 95–123.

[18] A. Djama, B. Djamaa, M.R. Senouci, Information-Centric Networking solutions for the Internet of Things: A systematic mapping review, Comput. Commun. 159 (2020) 37–59.

[19] L. Wang, A.K.M.M. Hoque, C. Yi, A. Alyyan, B. Zhang, OSPFN: An OSPF Based Routing Protocol for Named Data Networking, Technical Report NDN-0003, Name Data Networking, 2012.

[20] A.K.M.M. Hoque, S.O. Amin, A. Alyyan, B. Zhang, L. Zhang, L. Wang, NLSR: NAmed-data link state routing protocol, in: Proc. of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking, in: ICN '13, ACM, New York, NY, USA, 2013, pp. 15–20.

[21] P. Mendes, R.C. Sofia, J. Soares, V. Tsaoussidis, S. Diamantopoulos, C.-A. Sarros, Information-centric routing for opportunistic wireless networks, in: Proc. of the 5th ACM Conference on Information-Centric Networking, in: ICN '18, ACM, New York, NY, USA, 2018, pp. 194–195.

[22] X. Liu, Z. Li, P. Yang, Y. Dong, Information-centric mobile ad hoc networks and content routing: A survey, Ad Hoc Netw. 58 (2017) 255–268.

[23] L.M. J.S.M., V. Lokesh, G.C. Polyzos, Energy efficient context based forwarding strategy in named data networking of things, in: Proc. of the 3rd ACM Conference on Information-Centric Networking, ACM, New York, NY, USA, 2016, pp. 249–254.

[24] C. Yi, A. Afanasyev, I. Moiseenko, L. Wang, B. Zhang, L. Zhang, A case for stateful forwarding plane, Elsevier Comput. Commun. 36 (7) (2013) 779–791.

[25] C. Borrego, M. Amadeo, A. Molinaro, P. Mendes, R.C. Sofia, N. Magaia, J. Borrell, Forwarding in opportunistic information-centric networks: An optimal stopping approach, IEEE Commun. Mag. 58 (5) (2020) 56–61.

[26] G. Grassi, D. Pesavento, G. Pau, L. Zhang, S. Fdida, Navigo: Interest forwarding by geolocations in vehicular named data networking, in: IEEE 16th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015, pp. 1–10.

[27] M. Enguehard, R.E. Droms, D. Rossi, On the cost of geographic forwarding for information-centric things, IEEE Trans. Green Commun. Netw. 2 (4) (2018) 1150–1163.

[28] A. Rodrigues, P. Steenkiste, A. Aguiar, Analysis and improvement of name-based packet forwarding over flat id network architectures, in: Proc. of the 5th ACM Conference on Information-Centric Networking, in: ICN '18, ACM, New York, NY, USA, 2018, pp. 148–158.

[29] M. Antikainen, L. Wang, D. Trossen, A. Sathiaseelan, XBF: Scaling up bloom-filter-based source routing, 2016, CoRR abs:1602.05853. arXiv:1602.05853.

[30] J. Tapolcai, J. Bíró, P. Babarczi, A. Gulyás, Z. Heszberger, D. Trossen, Optimal false-positive-free bloom filter design for scalable multicast forwarding, IEEE/ACM Trans. Netw. 23 (6) (2015) 1832–1845.

[31] M. Amadeo, C. Campolo, A. Molinaro, Internet of things via named data networking: The support of push traffic, in: 2014 International Conference and Workshop on the Network of the Future (NOF), 2014, pp. 1–5.

[32] P. Moll, J. Janda, H. Hellwagner, Adaptive forwarding of persistent interests in named data networking, in: Proc. of the 4th ACM Conference on Information-Centric Networking, in: ICN '17, ACM, New York, NY, USA, 2017, pp. 180–181.

[33] C. Anastasiades, T. Braun, V.A. Siris, Information-centric networking in mobile and opportunistic networks, in: I. Ganchev, M. Curado, A. Kassler (Eds.), Wireless Networking for Moving Objects, Vol. 8611, Springer International Publishing, 2014, pp. 14–30.

[34] B. Nour, K. Sharif, F. Li, H. Moungla, A distributed ICN-based IoT network architecture: An ambient assisted living application case study, in: Proc. of IEEE GLOBECOM 2017, 2017, pp. 1–6.

[35] R. Ravindran, S. Lo, X. Zhang, G. Wang, Supporting seamless mobility in named data networking, in: Proc. of IEEE ICC, 2012, pp. 5854–5869.

[36] X. Vasilakos, V.A. Siris, G.C. Polyzos, M. Pomonis, Proactive selective neighbor caching for enhancing mobility support in information-centric networks, in: Proc. of the 2nd of the ACM ICN Workshop on Information-Centric Networking, 2012, pp. 61–66.

[37] F.R.C. Araújo, A.M. de Sousa, L.N. Sampaio, SCaN-Mob: An opportunistic caching strategy to support producer mobility in named data wireless networking, Elsevier Comput. Netw. 156 (2019) 62–74.

[38] G. Carofiglio, M. Gallo, L. Muscariello, ICP: Design and evaluation of an Interest control protocol for content-centric networking, in: Proc. IEEE INFOCOM Workshops, 2012, pp. 304–309.

[39] M. Amadeo, A. Molinaro, C. Campolo, M. Sifalakis, C. Tschudin, Transport layer design for named data wireless networking, in: Proc of IEEE INFOCOM Workshops, 2014, pp. 464–469.

[40] C. Anastasiades, L. von Rotz, T. Braun, Adaptive interest lifetimes for information-centric wireless multi-hop communication, in: 2015 8th IFIP Wireless and Mobile Networking Conference (WMNC), 2015, pp. 40–47.

[41] K. Matsuzono, H. Asaeda, C. Westphal, Network Coding for Content-Centric Networking/Named Data Networking: Considerations and Challenges, Internet-Draft draft-irtf-nwcrg-nwc-ccn-reqs-09, Internet Engineering Task Force, 2022.

[42] J. Saltarin, E. Bourtsoulatze, N. Thomos, T. Braun, NetCodCCN: A network coding approach for content-centric networks, in: IEEE INFOCOM 2016, 2016, pp. 1–9.

[43] R. Xu, H. Li, H. Zhang, Reliable content delivery in lossy named data networks based on network coding, in: J. Vaidya, J. Li (Eds.), Algorithms and Architectures for Parallel Processing, Springer International Publishing, Cham, 2018, pp. 34–46.

[44] CCN-Lite project, 2016, URL http://www.ccn-lite.net/#pub.

[45] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, T.C. Schmidt, RIOT OS: Towards an OS for the internet of things, in: 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2013, pp. 79–80, http://dx.doi.org/10.1109/INFCOMW.2013.6970748.

[46] A. Dunkels, B. Gronvall, T. Voigt, Contiki - a lightweight and flexible operating system for tiny networked sensors, in: 29th Annual IEEE International Conference on Local Computer Networks, 2004, pp. 455–462, http://dx.doi.org/10.1109/LCN.2004.38.

[47] E. Baccelli, C. Mehlis, O. Hahm, T.C. Schmidt, M. Wählisch, Information centric networking in the IoT: Experiments with NDN in the wild, in: Proc. of the 1st ACM Conference on Information-Centric Networking, 2014, pp. 77–86.

[48] B. Ahlgren, A. Lindgren, Y. Wu, Demo: Experimental feasibility study of CCN-lite on contiki motes for IoT data streams, in: Proc. of the 3rd ACM Conference on Information-Centric Networking, 2016, pp. 221–222.

[49] C. Gündoğan, P. Kietzmann, T.C. Schmidt, M. Wählisch, HoPP: Robust and resilient publish-subscribe for an information-centric internet of things, in: 2018 IEEE 43rd Conference on Local Computer Networks (LCN), 2018, pp. 331–334, http://dx.doi.org/10.1109/LCN.2018.8638030.

[50] C. Gündoğan, P. Kietzmann, M. Lenders, H. Petersen, T.C. Schmidt, M. Wählisch, NDN, CoAP, and MQTT: A comparative measurement study in the IoT, in: Proc. of the 5th ACM Conference on Information-Centric Networking, 2018, pp. 159–171.

[51] C. Boldrini, A. Passarella, HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships, Elsevier Comput. Commun. 33 (9) (2010) 1056–1074.

[52] A. Vahdat, D. Becker, Epidemic Routing for Partially-Connected Ad Hoc Networks, Tech. Rep. CS-200006, Duke University, 2000.

[53] E. Borgia, R. Bruno, A. Passarella, Reliable Data Delivery in IoT-ICN Environments, Technical Report, IIT TR-27/2020, IIT-CNR, 2020.

Raffaele Bruno received a Laurea (cum laude) degree and a Ph.D. in Computer Systems Engineering from the University of Pisa, Italy, in 1999 and 2003, respectively. He is currently a Senior Researcher at IIT-CNR, Pisa, Italy. He is the author of more than 90 peer-reviewed papers in international journal and conference proceedings. His current research interests include the design of architecture and protocols for IoT systems and 5G networks, fog/edge computing, and shared urban mobility systems. He was TPC co-chair of ACM Q2SWinet 2012, IFIP/IEEE SustainIT 2013, IEEE WoWMoM 2016, IFIP WWIC 2019, and general co-chair of IEEE WoWMoM 2021. He currently serves on the editorial board of Computer Communications (Elsevier) and Sustainable Computing (Elsevier), and he is Editor-in-Chief of the IoT section of Sensors (MDPI). He has been also guest editor of a number of journal special issues on topics related to wireless mesh networks, IoT and WLANs.

Eleonora Borgia is a Researcher at the Institute for Informatics and Telematics (IIT) of the National Research Council (CNR), Italy. She received the Laurea and the Ph.D. degree in Computer Engineering from the University of Pisa, Italy. She publishes her results on wireless and pervasive computing and cyber–physical systems topics in major international journals/conferences. She currently focuses on IoT systems, URLL communications and efficient data management at the network edge. She serves on the Editorial Board of Wireless Network (Springer) and International Journal of Distributed Sensor Networks (SAGE), and on the Technical Committee of Computer Communications (Elsevier), where she also was Guest CoEditor for a Special Issue on IoT. She was responsible for the scientific tasks/activities in European projects (e.g., EIT DIGITAL EIoT, IST-FP6 BIONETS). She was/is part of several organizing committee for international conference and workshop committees (e.g., IEEE GLOBECOM or IEEE ICC).

Andrea Passarella (Ph.D. 2005) is a Research Director at the Institute for Informatics and Telematics (IIT) of the National Research Council of Italy (CNR). Prior to join IIT he was with the Computer Laboratory of the University of Cambridge, UK. He has published 180+ papers on Online and Mobile social networks, decentralized AI, Next Generation Internet, opportunistic, ad hoc and sensor networks, receiving the best paper award at IFIP Networking 2011 and IEEE WoWMoM 2013. He served as General Chair for IEEE PerCom 2022. He is the founding Associate Editor-in-Chief of Elsevier Online Social Networks. He is co-author of the book "Online Social Networks: Human Cognitive Constraints in Facebook and Twitter Personal Graphs" (Elsevier, 2015), and was Guest Co-Editor of several special sections in ACM and Elsevier Journals. He is the PI of the EU CHIST-ERA SAI (Social Explainable AI) project.