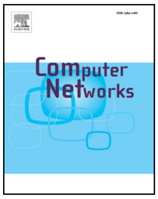
Computer Networks 000 (2016) 1–14

Contents lists available at ScienceDirect

Computer Networks

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



Information-centric content retrieval for delay-tolerant networks

Carlos Anastasiades*, Tobias Schmid, Jürg Weber, Torsten Braun

Institute of Computer Science, University of Bern, Neubrückstrasse 10, 3012 Bern, Switzerland

ARTICLE INFO

Article history:
Received 28 August 2015
Revised 15 March 2016
Accepted 18 March 2016
Available online xxx

Keywords:
Information-centric
Delay-tolerant
Agent-based content retrieval
Wireless routing
DTN
CCN

ABSTRACT

The shift from host-centric to information-centric networking (ICN) promises seamless communication in mobile networks. However, most existing works either consider well-connected networks with high node density or introduce modifications to ICN message processing for delay-tolerant networking (DTN). In this work, we present agent-based content retrieval, which provides information-centric DTN support as an application module without modifications to ICN message processing. This enables flexible interoperability in changing environments. If no content source can be found via wireless multi-hop routing, requesters may exploit the mobility of neighbor nodes (called agents) by delegating content retrieval to them. Agents that receive a delegation and move closer to content sources can retrieve data and return it back to requesters. We show that agent-based content retrieval may be even more efficient in scenarios where multi-hop communication is possible. Furthermore, we show that broadcast communication may not be necessarily the best option since dynamic unicast requests have little overhead and can better exploit short contact times between nodes (no broadcast delays required for duplicate suppression).

1. Introduction

The wide proliferation of embedded and mobile devices has driven recent advances for the Internet of Things (IoT) and pervasive mobile computing. In this area, information-centric networking (ICN) has been identified as a promising networking approach because it enables communication based on context (content availability in the neighborhood) [1–3]. ICN can support ubiquitous host mobility [4] because communication is not based on endpoint identifiers but on named data. Thus, content can be retrieved from the nearest node that can provide the content [5–10].

Existing routing protocols for wireless multi-hop communication consider either well-connected networks or delay-tolerant networks (DTN). In this paper, we investigate information-centric DTN support via agent-based content retrieval (ACR). In ACR, requesters can delegate content retrieval to mobile nodes (called agents) who will retrieve content on their behalf. Requesters can then retrieve content from agents when they meet them again. Earlier studies [11] have shown that human mobility exhibits temporal and spatial periodicity. This means that people tend to have strong location preferences in their daily mobility and meet other individuals regularly. ACR can exploit this property since agents (with different mobility patterns as requesters) can request content in locations requesters would never visit. In contrast to traditional opportunistic networking, no neighbor discovery is required. Users can broadcast requests and only nodes that can provide the desired content (or service) will reply. The lack of a reply means that no neighbor node can (or is willing) to provide the requested content, which—in terms of content retrieval—is equivalent to having no neighbors.

We base our work on Content-Centric Networking (CCN) [12], which is a popular ICN architecture that follows a layered structure. In CCN, routing and caching is performed on every node at a CCN daemon (CCND) and additional functionality can be provided by application modules (above the CCND). If applications provide a certain service or content, they can register prefixes at the CCND to receive requests from other nodes or applications. This means that the CCND includes only vital ICN functionality (to limit the processing complexity such that line-speed operations can be supported) and individual nodes can have extended functionality by running additional applications.

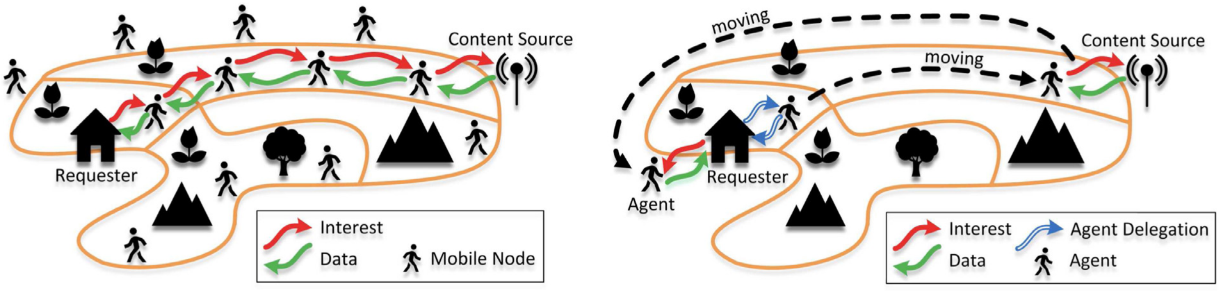
Consequently, we implement ACR as an application module to be run on mobile nodes. Since message processing at the CCND is not modified, ACR can be combined with multi-hop routing protocols such as Dynamic Unicast (DU, see Section 4), enabling interoperability in dense and sparse environments. For example, a requester could initially try to retrieve content via multi-hop DU if the node density is high (Fig. 1a). However, if the node density is low, content retrieval via multi-hop DU would not be

---

* Corresponding author. Tel.: +41 316318681.
E-mail addresses: anastasiades@inf.unibe.ch (C. Anastasiades), schmid@inf.unibe.ch (T. Schmid), weber@inf.unibe.ch (J. Weber), braun@inf.unibe.ch (T. Braun).

http://dx.doi.org/10.1016/j.comnet.2016.03.006
1389-1286/

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006



(a) Multi-hop Communication in Dense Environments (b) Agent-based Content Retrieval in Sparse Environments

Fig. 1. Content Retrieval in Different Environments.

successful. Then, requesters may delegate content retrieval to agents who move closer to content sources, can retrieve content and deliver it back to requesters as illustrated in Fig. 1b. Furthermore, requesters can switch back to multi-hop DU at any time to retrieve content from content sources or agents if the node density is sufficient to enable efficient multi-hop communication.

The contributions of this paper are as follows:

- We present agent-based content retrieval (ACR), which provides information-centric DTN support as an application module without modifications to ICN message processing. This enables interoperability between dense and sparse networks, i.e., the combination of DTN and multi-hop routing.
- We implement ACR and compare it to multi-hop routing in extensive evaluations. We show that ICN communication can benefit from mobility such that ACR is superior to multi-hop routing (message overhead and content retrieval time) even in cases where multi-hop routing is possible.
- We show that broadcast communication is not necessarily the best option in mobile multi-hop communication. Dynamic Unicast (DU) has little overhead and is robust in low mobility scenarios as well as high mobility scenarios with limited path length. Breaking of symmetric (Interest-Data) forwarding paths is no issue because content returns within milliseconds, i.e., the topology has not changed much.

To increase the credibility of our results, we have implemented all protocols in CCNx 0.8.2 [13] and evaluated them by extensive evaluations with NS3-DCE [14] on a Linux cluster [15].

The remainder of this paper is organized as follows. In Section 2 we give an overview of CCN and multi-hop content retrieval. Section 3 describes our design for ACR. Multi-hop content retrieval for dense environments is described in Section 4. Evaluation results are presented in Section 5 and we describe our lessons learned in Section 6. Finally, we conclude our work in Section 7.

2. Related work

2.1. Content-centric networking

Content-Centric Networking (CCN) [12] is based on two message types: *Interests* and *Data*. Content is organized in segments. Every segment, which is included in a Data message, has a unique name and is signed by the publisher who created the content. CCN follows a pull-based communication strategy, where requesters need to send Interests for all segments of a content object to retrieve the complete content object. Interests are forwarded based on longest-prefix matching and Data follows the reverse path back to requesters. The CCNx project [13] provides a reference implementation of the CCN concepts. The core element of CCNx is the CCN Daemon (CCND), which performs message processing and forwarding decisions. Links between forwarding engines (CCNDs) on different hosts or local applications are called faces.

CCN message processing is based on a Forwarding Information Base (FIB) to forward Interests towards content sources, a Pending Interest Table (PIT) to aggregate already sent Interests for the duration of an Interest lifetime (or less if satisfied by Data earlier) as well as a Content Store (CS), which is used as cache. Content is persistently stored and provided to others by repositories, which are implemented as applications. Repositories are linked to the CCND via internal faces.

Named Data Networking (NDN) [16] is a NSF-funded Future Internet project, which was originally based on CCNx. Although NDN and CCN have split in August 2013, NDN continued using a modified CCNx implementation (renamed to NDNx [17]). Since August 2014, NDNx features a new implementation including a new NDN Forwarding Daemon (NFD). In this paper, we used CCNx 0.8.2, because it was the latest available CCN source code at the time of our evaluations. However, with the publication of CCNx 1.0 [18], there have been major changes to CCNx, e.g., longest-prefix matching (LPM) and Exclude filters (EFs) [19] are no longer supported, while they are still supported in NDNx [20]. Since LPM and EFs are required to create an agent list for agent delegation, future ACR implementations should rather be based on NDNx instead of CCNx. Despite the new implementation, NDNx concepts and message fields are still similar to previous CCNx 0.x versions, which we used in our evaluations.

2.2. Multi-hop content retrieval

Multi-hop content retrieval is required if a requester can not reach a content source directly. We differentiate between wireless routing protocols in connected networks and delay-tolerant communication protocols in disrupted networks.

2.2.1. Wireless ICN routing protocols

Many wireless ICN routing protocols rely exclusively on broadcast communication. It has been shown [6] that ICN outperforms MobileIP for vehicular communication with high node velocities. Several enhancements [7] have been proposed for wireless (multi-hop) broadcast communication such as a collision avoidance timer (duplicate suppression) and a pushing timer (shorter forwarding delays for vehicles farther away from the content source). While location information encoded in names [7] may enable traffic information to be quickly forwarded further away from where it was originated, it would also require forwarding strategies to understand name semantics.

To avoid complex forwarding strategies based on name semantics, a Link Adaptation Layer (LAL) [21] has been proposed to support broadcast forwarding based on distance information. It is a 2.5 layer protocol that appends GPS coordinates in additional headers to all MAC messages. Besides a GPS receiver, every node

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

requires a digital map to locate nodes on the map. Messages are forwarded with a delay based on the distance to the previous node: the shorter is the distance, the longer is the wait. More recent work [10] built upon LAL has introduced the concept of geo-faces. Data names of producers are bound to geographic regions (squares on a digital map) such that Interests can be directed towards these regions. The location (region) of Data producers is discovered via flooding. The most evident drawbacks of these approaches are the significantly increased processing overhead to compare coordinates in all messages (location-based forwarding) and the requirement to use GPS receivers and digital maps. Furthermore, squares on a digital map may be traveled quickly, i.e., particularly vehicles near borders may change regions frequently, resulting in imprecise routing decisions.

Another approach for pedestrian mobility uses hop distance information for forwarding [8]. To support this, hop information and provider IDs are included in Interest and Data messages. While this approach may run on resource constrained devices, it introduces other significant drawbacks. First, if provider IDs are added to messages, communication is not strictly information-centric anymore since provider IDs matter such that provider handovers [8] are required. In addition, Interest aggregation in the PIT becomes more complex for multiple requesters because of different provider IDs and (potentially) different hop distances. Furthermore, hop distances may be inaccurate for cached Data in case of mobility. If hop distances and provider IDs would be updated at every hop, they cannot be protected by the publisher's signature making it vulnerable to attacks.

To retrieve content quickly from any available content source and enable resilience to user mobility, all approaches described above are based on broadcast exclusively. This means that Interest messages may address multiple content sources at the same time. With ubiquitous caching in ICN, content density increases drastically and even more nodes may be able to return Data in response to broadcast Interests. To avoid collisions and enable duplicate suppression, Data messages need to be delayed. However, these broadcast delays decrease broadcast throughput significantly. In earlier work we have, therefore, developed Dynamic Unicast (DU) [22] for opportunistic one-hop communication. DU uses broadcast communication only to find a content source but as soon as Data is returning back, a dynamic unicast face can be configured towards the content source. Dynamic Unicast can also be extended to wireless multi-hop communication and we describe this in Section 4.

Experiments in a static wireless sensor test bed [2] have shown that dynamically configuring the FIB with unicast routes to the next hop, i.e., Reactive Optimistic Name-based Routing (RONR), can reduce radio transmissions during multi-hop communication by 50% compared to broadcast. However, neighbors may change in mobile networks and it remains open whether dynamically created unicast faces are also efficient in mobile networks.

2.2.2. Delay-tolerant communication protocols

Routing in delay-tolerant networks (DTN) [23] has been studied for more than a decade. Many DTN routing protocols [24] have been proposed such as Epidemic Routing [25], where a node copies its messages to all nodes that it encounters, Spray-and-Wait [26], which limits the number of forwarded copies, or prediction-based forwarding based on the history of past encounters [27]. In recent years, increasingly more DTN routing protocols rely on social characteristics [28] to improve message delivery. Based on neighbor discovery [29], nodes can create a contact graph to detect communities as well as extract centrality, similarity and friendship characteristics for more efficient forwarding. However, mapping contacts to social relations is complex and DTN routing performance heavily depends on how the mapping (contact aggregation) is performed [30].

By targeting named data rather than node endpoints, ICN can support efficient DTN communication enabling requesters to retrieve desired content quickly from any neighboring device (if the content is available). CEDO [31] is a first approach to extend CCNx with DTN functionality. Interests stay in the PIT until they are satisfied. Whenever a contact is detected, a message that summarizes all pending Interests is transmitted. A receiver of such a message sends back all Data messages that it has in the cache. Regular Interest forwarding via FIB is disabled. Another approach [32] introduces the concept of logical faces for communication in disaster scenarios. It assumes that communities (location where content can be retrieved and from where Interests have originated) are static and mobile mules forward messages between communities. Logical faces are used to map communities to physical interfaces. Whenever mules reach communities, the corresponding logical faces become active and Data can be retrieved based on PIT information.

Both approaches [31,32] keep Interest messages in the PIT until nodes encounter the desired content source or community. However, CCN content may contain multiple segments, which need to be requested individually and, typically, the requester does not know the content size until receiving the last segment. Thus, a requester does not know how many Interests are required to retrieve the content and there is no immediate feedback when content transfer is finished (delay-tolerant retrieval via data mules). If too few Interests are forwarded by a requester, a data mule may not retrieve the complete content and may need to travel several times between communities. Too many Interests, however, result in inefficient PIT memory management. Since both approaches [31,32] modify CCN message structures and message processing at the CCND, they are incompatible to CCN routing protocols for well-connected networks.

Traditional opportunistic and delay-tolerant networking approaches use neighbor discovery [29,33] to detect neighbor devices before communication takes place. However, from an information-centric perspective, it may be questionable whether device discovery is required at all. The existence of neighbor devices does not reveal any information about available content or a neighbor's ability and willingness to perform certain tasks. Agent-based content retrieval (ACR) enables information-centric delay-tolerant communication as an application module. Then, the decision when to forward Interests in sparse environments is provided by the application module enabling more flexible application-specific connection criteria. A node could transmit a request when it is interested in a content object or service, and a neighbor node would only answer (become a contact) if it can satisfy the request.

An initial design for ACR has been described and evaluated in a small static testbed of three Android smart phones and a laptop [34]. In this work, we refine agent delegation and content notification mechanisms for disrupted interactions, enable flexibility (and extensibility) by optional parameters and adapt message naming to follow the CCNx Basic Name Conventions [35] (see Section 3). Furthermore, we integrate ACR with DU [22] and compare it to multi-hop routing in mobile networks with up to 100 mobile nodes.

3. Agent-based content retrieval for delay-tolerant communication

In ACR, requesters can delegate content retrieval to agent nodes. In most cases, the exact content size may not be known. Hence, it is impossible to transfer a sufficient and not extensive amount of Interests (for every content segment) to an agent (see Section 2.2.2). However, this is not required since only the task to retrieve content is transferred. Agents are implemented as application modules and can perform content retrieval independently. Later, after receiving a notification from the agent, the requester

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

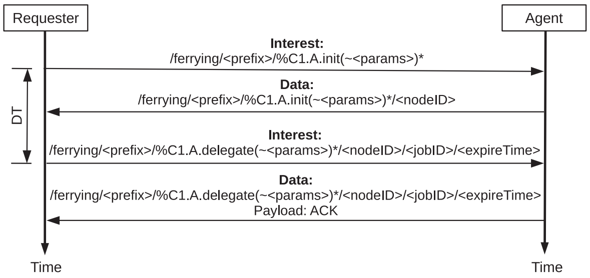


Fig. 2. Message Sequence during Agent Delegation.

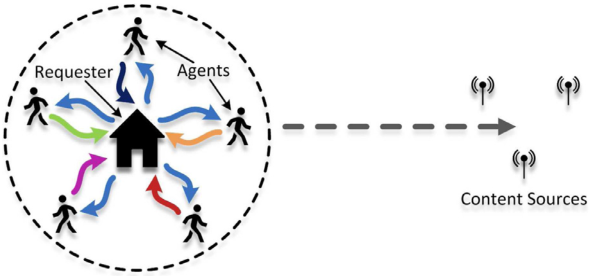


Fig. 3. Agent Delegation: a broadcast Exploration Interest from a requester may retrieve multiple Exploration Data replies from different agents.

can regularly retrieve the content from the agent via multiple Interests. ACR is performed in three phases, which are described below.

3.1. Phase I: Agent delegation

Agent delegation describes the process of finding an agent and delegating content retrieval to it. The proposed agent delegation sequence is presented in Fig. 2. A requester broadcasts an Exploration Interest with the prefix */ferrying*, the content name <*prefix*> and optional selection parameters, e.g., coordinates where the content may be found, to its one-hop neighbors. Agents have registered Interest filters for the prefix */ferrying* at the CCND to receive Exploration Interests. If agents have sufficient resources to perform the task and agree with the optional parameters, they reply to an Exploration Interest with Exploration Data appending their nodeID, which uniquely identifies the agent, in the name. Since the Exploration Interest is broadcast (see Fig. 3), the requester may receive multiple Exploration Data replies from agents in one-hop distance.

The requester can then create an agent list by retrieving other replies from its local cache (using Exploration Interests with Exclude filters [19] that exclude already known nodeIDs). Exploration Data has a short lifetime of only a few seconds to avoid usage of old information from the cache. After a short delegation time (DT, in our implementation: 2s), where additional responses may be retrieved from the cache, the requester can select an agent from the agent list for delegation. Agent selection can be based on diverse criteria such as social relations or past GPS traces. For example, if a requester knows at which location or in which area desired content can be found, it can indicate this as optional parameter in Exploration Interests such that only agents who travel in this area may respond. Since most smart phones nowadays store GPS traces locally, an agent can simply examine it's locally collected GPS traces without requiring transmitting it or sharing it with others. However, for the sake of simplicity, we randomly select agents from the agent list in this paper and leave more sophisticated criteria as well as multi-hop agent selection algorithms for future work.

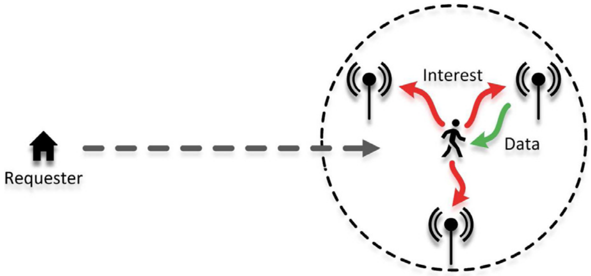


Fig. 4. Content Retrieval by agents: broadcast requests enable implicit content discovery, i.e. agents can find the content quickly at any neighbor node.

The requester sends a Delegation Interest to the selected agent using its nodeID, a jobID, an expiration time and optional parameters such as the notification type *push* or *pull* (previous work [34] used only pull notifications). The jobID is used in the notification phase (see below) and the expiration time limits the duration that an agent is looking for the content. As a last step, the agent has to confirm the delegation with an acknowledgment (ACK). The last step is required to ensure that the agent has received the delegation and has not moved away, i.e., without acknowledgment (as in previous work [34]) a requester may wait infinitely long for content if the agent has never received the delegation. Agent delegation can be performed to one or multiple redundant agents up to an agent limit (maximum number of delegated agents).

3.2. Phase II: Content retrieval

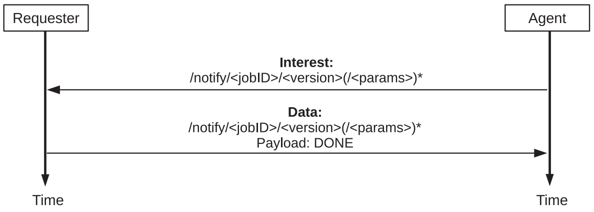
After agent delegation, the agent can find and retrieve content for the requester by periodic Interests (Interest probing, in our implementation: every 1s). Since neighbor nodes may change, static unicast faces to content sources cannot be configured and communication needs to be performed via broadcast. Broadcast requests enable implicit content discovery [36] as illustrated in Fig. 4, i.e., a broadcast request can address multiple nodes at the same time but only a content source, which holds the desired content, will reply. Content retrieval can also be performed via Dynamic Unicast [22], where content requests are transmitted via broadcast only until a content source is found. Then, subsequent Interests are addressed via unicast to the same content source until it becomes unavailable. Although content retrieval can be performed via opportunistic one-hop or multi-hop communication, we limit the scope of ACR requests in our current implementation to one-hop and leave evaluations of multi-hop for future work.

Content retrieval is performed in two steps. First, the agent resolves the content version. Second, as soon as a version has been found, content retrieval can start. To persistently store the content and keep the publisher's original signatures, the agent delegates content retrieval to it's local repository, which is an independent application that retrieves and persistently stores desired content. Interactions between local applications and repositories (via internal faces through the CCND) follow the repository protocol [37]. We have extended the repository protocol i) to check whether the content has been retrieved completely and ii) to resume incomplete content retrievals from where they have stopped in case of disruptions. When content retrieval is complete, the notification phase starts.

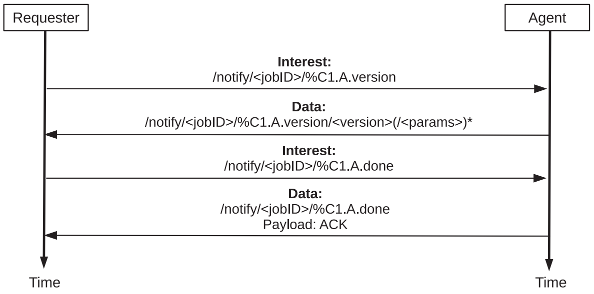
3.3. Phase III: Content notification

When an agent has retrieved the content, it can notify the requester via push or pull notifications. Then, after receiving a notification, the requester can retrieve the content from the agent. The decision which notification type to use is made by the requester

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006



(a) Push Notification



(b) Pull Notification

Fig. 5. Message Sequence for Push and Pull Notification.

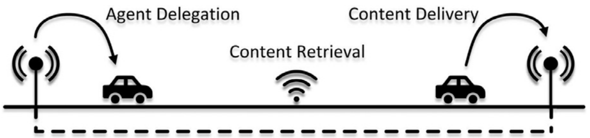


Fig. 6. Agent Delegation and Content Delivery through access points at different locations: the access points can coordinate each other via wired link (dashed line).

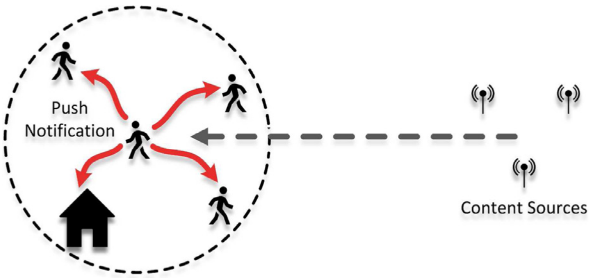


Fig. 7. Push Notifications by agents: after reception, requesters can start content downloads.

during agent delegation. Both notification types are based on the assumption that agents meet requesters again after a while. However, agent delegation and content delivery can also be at different locations as long as both locations can communicate and coordinate with each other. For example, Fig. 6 shows two access points, which are placed along a road. The access points are connected to each other via wired link (dashed line). Then, if an access points can not cover the entire road, agent delegation may be performed at one access point and content delivery may be performed at another access point further down the road, while agents can collect information in-between.

3.3.1. Push notification

As soon as an agent has retrieved the content, it can start the notification phase by periodically transmitting push notifications (see Fig. 7). When the requester receives the push notification, it can start retrieving the content from the agent.

Fig. 5a illustrates the message sequence for push notifications. Push notifications are Interest messages with the prefix */notify*, the *jobID* (see agent delegation) and the content version. Furthermore, agents may add additional parameters for content retrieval, e.g., their nodeIDs. In our implementation, we use IP addresses as nodeIDs but also other node identifiers, e.g., MAC addresses or descriptive names, are possible. NodeIDs are required such that requesters can create direct unicast faces to agents to retrieve content with a higher throughput, i.e., there are no broadcast delays during unicast communication and rate adaptation mechanisms are supported. Thus, in contrast to [8], nodeIDs are only used to create unicast faces to neighbor nodes but they are not included into CCN messages. As soon as content retrieval has finished, the requester notifies the agent (*DONE* flag) indicating that no more notifications are required.

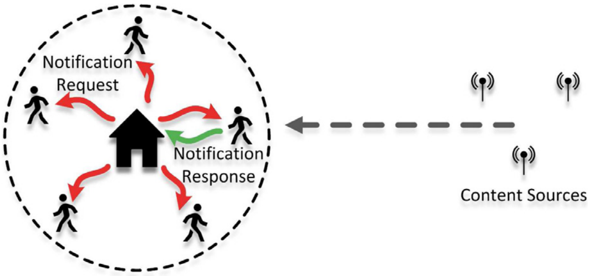


Fig. 8. Pull Notifications by requesters: agents reply with Notification Responses if they have retrieved the content.

3.3.2. Pull notification

Pull notifications are based on periodic Notification Requests transmitted by requesters followed by Notification Responses transmitted by agents if they have retrieved the content (see Fig. 8). The message sequence is shown in Fig. 5b. Agents that have completed content retrieval can register an Interest filter in the jobID to receive Notification Requests, i.e., Interests for the jobID. Then, as soon as an agent comes into the requester's transmission range and receives the Notification Request, it can respond with a Notification Response containing the content version and optionally the nodeID (for direct content retrieval similar to push notifications). Since multiple agents may be delegated with the same jobID for redundancy, a requester can retrieve notifications from any agent in its neighborhood with one message.

After a requester has finished content retrieval from an agent, it can notify the agent (via extensible command marker [35] *%C1.A.done*) to delete the job and release the resources for other jobs.

Please note that push notifications have a larger size than pull notifications (in our implementation push notifications are around 35 bytes larger) because they contain all information to retrieve content (e.g., nodeID, content version). Pull notifications can be short because additional information is only transmitted if requester and agent meet.

4. Dynamic unicast for multi-hop content retrieval

ACR can be combined with multi-hop routing. For example, requesters could find agents or agents could find content sources, which are multiple hops away. Furthermore, ACR may not be required at all in dense environments because requesters can retrieve content directly via multi-hop communication. In this section, we describe multi-hop DU, which is an extension to one-hop DU [22] and an alternative to multi-hop broadcast communication.

4.1. Prefix registration

Requesters can find agents or content sources by flooding Interests for the desired content. To enable flooding for arbitrary content names, we slightly modified CCNx such that Interests are

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

transmitted via a broadcast face if there is no FIB entry configured. Agents or content sources answer broadcast Interests with broadcast Data responses such that other sources can detect and suppress duplicate transmissions. Then, unicast FIB forwarding entries can be configured by Data messages on the reverse path (content name without segment number and IP address of the previous hop as nodeID).

4.2. Forwarding strategies

To retrieve content over multiple hops, we define the following two FIB forwarding strategies.

4.2.1. Single face forwarding (SFF)

Every Interest is first forwarded over the “best” face and if nothing has been received, it is forwarded via broadcast (fallback). If a unicast face is available, it is the best face (priority over broadcast). If multiple unicast faces are available, the face that was (successfully) used for the previous Interest is considered the best face.

4.2.2. Parallel faces forwarding (PFF)

Similar to SFF, Interests are forwarded first via unicast faces. Different from SFF, Interests are always forwarded over all available unicast faces in parallel and not only over one (“the best”) face. If nothing has been received in return, the Interest is sent via broadcast (fallback).

4.3. Route updates and path breaks

Unicast routes may break quickly in mobile multi-hop environments since neighboring nodes may see each other only for a short time. Since Interests transmitted over broken paths increase message overhead and transmission times (if alternative paths exist), expired forwarding information needs to be removed quickly. Therefore, we set the lifetime of dynamically configured FIB entries to only five seconds, i.e., slightly more than the default Interest lifetime of four seconds. By that, retransmissions (in case of collisions) can still be satisfied from nearby caches while broken paths expire quickly. Whenever a Data message is received over a configured face, the lifetime of the corresponding FIB entry is extended. If the configured FIB entry expires, another Interest needs to be broadcast according to the fallback strategy. This enables nodes to establish new paths to potentially closer content sources.

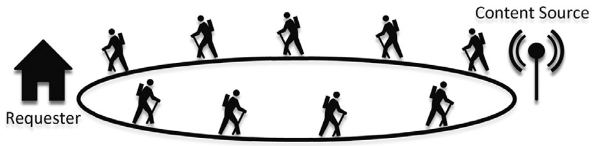
Thus, multi-hop DU can exploit all three CCN memory components of the CCND. First, Interests are forwarded based on dynamically configured FIB entries depending on content source availability (flexibility). Second, Interests from multiple requesters can be merged and aggregated in the PIT of forwarder nodes such that only a fraction of all Interests need to be forwarded to a content source (scalability). Third, received Data messages are kept in the cache for a short time such that retransmissions (in case of collisions) can be satisfied from intermediate caches (reliability).

5. Evaluation

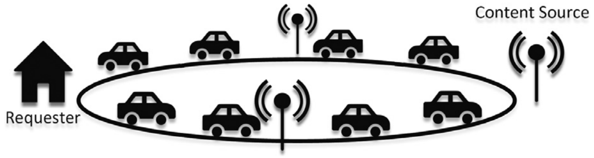
We have implemented agent-based content retrieval (ACR) and Dynamic Unicast (DU) in CCNx 0.8.2 [13] and compared it to multi-hop broadcast communication using standard CCNx (as reference since dynamically created unicast paths may have only a limited lifetime in mobile scenarios). To enable multi-hop broadcast communication, we have configured two broadcast faces that can be used for Interest forwarding in alternating order as described in [38]. By that, a received Interest on one face can be forwarded (at most once) on the other face if it has not yet been received by another node before (no existing PIT entry). The evaluation has been performed with NS3-DCE [14] on a Linux cluster [15]. By that, we deploy the same source code on simulated nodes that would run on real mobile devices. Although this evaluation method introduces limitations in terms of network size and simulation time, we believe that it increases accuracy and practical relevance of our evaluations.

Table 1
Evaluation Parameters.

Parameter	Value
Wireless standard	IEEE 802.11g, 2.4GHz
Modulation	ERP-OFDM, min. data rate: 6Mbps max. data rate: 54Mbps
Propagation loss model	Log distance with exponent: 3.0 Reference loss: 40.0dB
Energy detection threshold	−86.0dBm
CCA Model threshold	−90.0dBm
Mobility	circular mobility regular: 1.0–1.4m/s, node pause: 0-30s slow: 0.7 -1.0m/s, node pause: 0-1200s fast: 10.0–14.0m/s, node pause: 0-30s
Circle radius	250m (circumference: 1570.8m) 375m (circumference: 2356.19m) 500m (circumference: 3141.59m)
Nodes	1 static requester, 1–3 static sources mobile nodes: 5–100 agent limit: 1, 5, 10 agent delegation: every 10s up to limit
File sizes	0.5MB, 1MB, 5MB, 10MB, 20MB segment size: 4096 bytes



(a) hiker scenario



(b) vehicular scenario

Fig. 9. Evaluation Topologies.

5.1. Scenarios and configuration

The evaluation parameters are listed in Table 1. Every node has an IEEE 802.11g wireless interface and we use a log distance propagation loss model. With the selected parameters, the transmission range is approximately 130 m (outdoor scenario). The data rate is adapted automatically based on the distance between two nodes, i.e., the signal-to-noise ratio (SNR).

Human mobility follows cycles among different points of interest, e.g., work and home, by traveling on urban roads or using public transportation systems. In our evaluations, we consider circular mobility, where (CCN capable) nodes move with different velocities on a circle. Fig. 9 shows the evaluation topologies for a hiker and a vehicular scenario. Mobile nodes are randomly distributed on the circle and move with velocities within a specified range depending on the scenarios described below. Node pauses specify individual occasional breaks (randomly selected within the specified intervals in Table 1), where nodes do not move. Although mobile nodes return to the requester in our evaluations, this is not neces-

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

sarily required as described in Section 3.3. For ACR, every mobile node is an agent. Requesters can delegate content retrieval to mobile nodes (agents), at best effort every 10s until the agent limit, i.e., the maximum number of delegated agents, has been reached. If there is no agent available due to low node density, agent delegation is postponed by five seconds.

Since end-to-end paths (for multi-hop communication) can be disrupted, we use a content retrieval application [39] that persistently stores received segments at requesters. Then, even in case of long disruptions (when cached content may be deleted), content downloads can always be resumed by requesters from where they were stopped. Different from agents, requesters do not need to provide persistently stored content to others but can store it privately (not in the repository). Still, received Data messages are stored temporarily in caches of intermediate nodes enabling quick retransmissions.

Every configuration has been evaluated in 100 different simulation runs.

5.1.1. Hiker scenario

In the hiker scenario (Fig. 9a), we place a static requester and a static content source on opposite sides of the circle to ensure multi-hop communication. Consider a round trip hiking trail to the top of a mountain. At the top, there is a content source, e.g., a solar powered sensor node that is gathering data (e.g., weather data, web cam snapshot etc.). At the start of the trail there is the tourist office, which is interested in the sensor data. Since there is no direct connection between tourist office and sensor, multi-hop routing or ACR needs to be applied. Hikers travel with regular pedestrian speeds of 1.0–1.4 m/s on the trail and make a short break from time to time to enjoy the view or take a picture, i.e., node pause times of 0–30 s. In addition, there are lazy hikers, who travel with slower speeds of 0.7–1.0 m/s and make longer breaks between 0s and 1200s.

5.1.2. Vehicular scenario

The vehicular scenario (Fig. 9b) is similar to the hiker scenario but nodes move with vehicular speeds of 10.0–14.0 m/s resulting in significantly shorter contact times between nodes. Therefore, two additional redundant content sources are placed on the circle, e.g., access points that are connected to the Internet, such that requester and content sources are in equidistance (1/4 circumference) to each other. Then, if content retrieval from one content source is disrupted, it can be resumed from another content source. Consider for example travelers on a safari in a wildlife park. While moving in their cars, travelers may collect sensor information from their surroundings (e.g., images from animal surveillance cameras). From time to time, travelers make short stops to watch animals more closely and they can deliver collected information at the exit when leaving the park. Other mobility examples may include public transportation systems or mail delivery services in urban areas, where users follow a specific route and then return back.

5.2. Push vs. pull notifications

We first evaluate the notification types for ACR with one-hop broadcast in the hiker scenario. Fig. 10 shows the notification messages (y-axis) transmitted via push and pull notifications when all nodes move with regular speeds (1.0–1.4 m/s) on a circle with radius 250 m. We evaluate the performance for different numbers of nodes (x-axis) in the network resulting in different node densities.

Fig. 10 shows that the number of pull notifications stays approximately constant independent of the number of delegated agents (agent limit) because one pull request can retrieve content from any agent node in the vicinity. However, more push notifications are required if the number of delegated agents increases because each agent transmits them individually. Furthermore, the number of push notifications increases with more mobile nodes (agents) in the network because agents can be delegated faster, i.e., agents can be delegated shortly after each other such that they start their notification phases approximately at the same time. If fewer mobile nodes are available, agent delegation takes more time, i.e., some agents may have already returned the content to the requester before other agents have retrieved the content and started their notification phases.

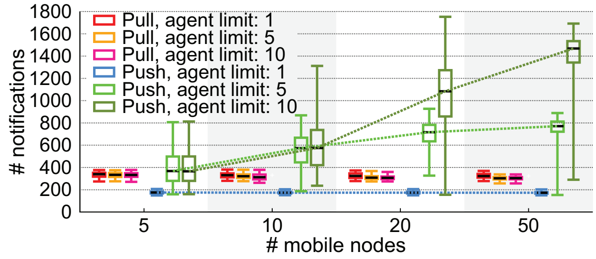


Fig. 10. Number of Push and Pull Notification Messages for a Varying Number of Mobile Nodes.

For 10 delegated agents out of 50 mobile nodes, push notifications result on average in 3.8 times more notification messages than pull notifications. However, if there is only one delegated agent, pull notifications result in 87% more messages than push notifications. This is because we start pull notification requests immediately after delegating the first agent (no assumptions when content is retrieved), while push notifications start only after the agent has retrieved the content. Thus, pull notifications are transmitted for a longer time. However, optimizations for pull notifications are possible by estimating the time for agents to retrieve content and return back. As a rule of thumb, we can say that pull notifications should be used if content retrieval is delegated to more than one agent (fewer notifications) and push notifications are more efficient for delegations to one agent (since notifications start only after the agent has retrieved the content).

Please recall that pull notifications are smaller because additional information is only transmitted if requester and agent meet while push notifications contain all information to retrieve content (see Section 3.3). When considering the sizes of transmitted messages for one agent, pull notifications result in only 25% (or 5.5 KB) more traffic (not optimized case). Thus, if notification messages need to be transmitted periodically, pull notifications are favorable in terms of message size compared to push notifications. In the remainder of this paper, we only use pull notifications.

5.3. Agent-based vs. multi-hop content retrieval

We compare ACR with one-hop broadcast requests against multi-hop communication (broadcast and DU with SFF or PFF), on a circular topology with a radius of 250 m. For that, we define the time until a requester has retrieved desired content as *content retrieval time*. Fig. 11 shows content retrieval times for a 1MB file (y-axis) for different numbers of nodes in the network (x-axis). In this scenario, 75% of the nodes move with slow speeds and make long breaks and only 25% of the nodes move with regular speeds (hiker scenario, cf. Table 1). The horizontal area between the dotted lines denotes the min./max. traveling times of an agent around the circle (regular speed, no pause time).

Fig. 11 illustrates that ACR retrieval times mostly depend on the nodes' mobility characteristics while multi-hop communication depends on the number of nodes (density) in the network. If con-

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

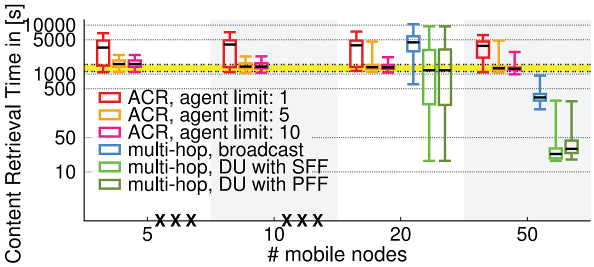


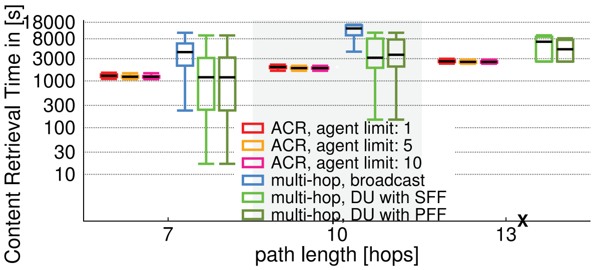
Fig. 11. Content Retrieval Times of a Requester on a Circle Topology with 250m Radius, 75% Slow Nodes and 25% Regular Nodes.

tent retrieval is delegated to only one agent, there is a high risk that it is a slow node (since most nodes are slow in this scenario) resulting in long content retrieval times. However, by delegating content retrieval to multiple redundant agents, i.e., in this scenario five agents are enough, the impact of slow nodes becomes negligible. Furthermore, we can observe that ACR is successful with any number of nodes while multi-hop communication is only possible for 20 and 50 nodes.

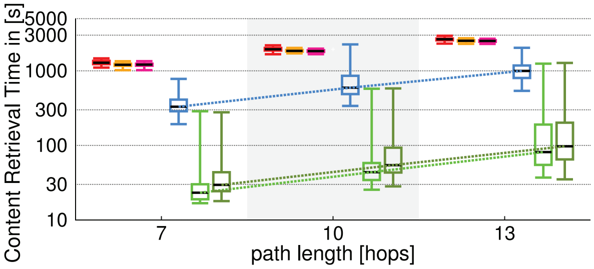
Scenarios with 20 nodes correspond to an average distance of 78.5 m between nodes, which we define as *intermediate node density*, and scenarios with 50 nodes correspond to an average distance of 31.4 m between nodes, which we define as *high node density*. Although communication with both node densities is expected to work well (transmission range of 130 m), this is not the case as Fig. 11 shows. For 20 nodes, ACR with five agents results in 59% (median) shorter content retrieval times than multi-hop broadcast and requires only 13% (median) more time than DU with SFF or PFF. However, multi-hop communication experiences a large variability for this node density. If nodes are favorably clustered between requester and content source, multi-hop communication performs better than ACR while it performs worse if this is not the case (long disruption periods). For 50 nodes, the node density is high enough such that multi-hop communication is always faster than ACR. In this case, multi-hop broadcast results in 2.9 times faster transmission than ACR (with five agents) and multi-hop DU in even 44 times (PFF) or 55 times (SFF) faster transmission than ACR.

Furthermore, we can observe that multi-hop DU (with SFF or PFF) performs significantly better than multi-hop broadcast. This is due to two main reasons. First, since broadcast requests are addressed to all nodes of a node's vicinity, broadcast transmissions need to be delayed to enable duplicate suppression. In multi-hop communication, these delays have a significant impact on throughput because they are added at every hop. In our evaluations, we used the default broadcast delay (CCNx data pause) of 10 ms, and larger values resulted in significantly worse performance. Only broadcast communication requires these delays since unicast requests address nodes directly. Second, the data rate for unicast transmissions can be adapted dynamically based on the distance between two nodes, i.e., based on the signal-to-noise ratio (SNR). For broadcast communication, however, the data rate can not be adapted and is usually set to the lowest supported rate (see Table 1). Although only a few nodes receive content via unicast, multi-hop DU can still exploit caching, i.e., if a multi-hop path breaks, Interests do not need to be retransmitted over the entire path (in most of the cases) since Data can be retrieved from a mobile node's cache, i.e., where the path broke.

Fig. 11 shows that ACR can be combined with multi-hop DU, e.g., a requester could initially try to retrieve content via multi-hop communication and only switch to ACR if nothing can be received. In very sparse or very dense environments, the combination is straightforward. For example, for 5 or 10 nodes in Fig. 11, multi-hop DU does not work and a requester could switch to ACR after a few expired content requests. Similarly, for 50 nodes a requester could directly retrieve the content via multi-hop DU such that ACR is not required. However, in dynamic and time-varying environments, i.e., neither permanently dense nor sparse, the combination becomes more complex and requires further investigation. In particular, it needs to be studied how long a requester should try to find an alternative path in case of a disruption before content retrieval is delegated to an agent.



(a) Intermediate Node Density



(b) High Node Density

Fig. 12. Content Retrieval Times of a Requester for Varying Path Lengths and Different Node Densities.

5.4. Agent-based vs. multi-hop content retrieval for higher node densities

In this section, we compare ACR with one-hop broadcast requests to multi-hop communication for increasing path lengths with intermediate and high node densities. All nodes are moving with regular speeds of 1.0–1.4 m/s (hiker scenario).

5.4.1. Content retrieval times

Fig. 12 shows content retrieval times for a 1MB file (y-axis) for both node densities and varying path lengths (x-axis). The path length indicates the minimum number of required hops between requester and content source, i.e., seven hops for a circle radius of 250 m, 10 hops for a circle radius of 375 m and 13 hops for a circle radius of 500 m. To keep the same node densities, we increased the number of mobile nodes accordingly, e.g., from 50 to 75 nodes (375 m radius) and to 100 nodes (500 m radius).

Fig. 12a shows content retrieval times for the intermediate node density. For seven hops, ACR performs better than multi-hop broadcast and similar to multi-hop DU (median values) as seen in the previous subsection. However, with increasing path length, multi-hop communication becomes significantly worse. For 13 hops, multi-hop content retrieval via broadcast can not be completed during six hours (our maximum simulation time), i.e., in 77% of the simulation runs requesters retrieve up to 11% of the content, while in 23% of the runs they do not retrieve a single segment. The performance is only slightly better with multi-hop DU because continuous end-to-end paths are only available for a short time, i.e., only in 5% of the runs requesters could retrieve the complete content. The PFF strategy results in slightly shorter content

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

retrieval times than SFF for long path lengths due to better path redundancy. However, for intermediate node densities, multi-hop communication with more than seven hops is barely successful or requires a lot of time. Thus, ACR should be preferred for long path lengths in such scenarios.

Fig. 12b shows content retrieval times for the high node density. Even for long path lengths, multi-hop communication is always successful and significantly faster than ACR. This is mainly due to the fact that end-to-end path availability is more stable and agents need to travel the entire path before delivering content. We can confirm path stability by observing the number of path breaks and resume operations. For 13 hops with high node density (Fig. 12b), approximately five resume operations were necessary on average with broadcast, while for the intermediate node density (Fig. 12a) already 25 resume operations were required for seven hops and more than 100 resume operations for 10 hops (a 1MB file has 251 segments). Strategies to combine ACR with DU should, therefore, not only depend on mobility patterns, e.g., node velocity and density, but also on the number of disruptions in relation to the size of already received partial files.

5.4.2. Transmitted messages

Multi-hop DU establishes a path between requester and content source such that only nodes on the path receive and forward messages, while for multi-hop broadcast all nodes receive messages and decide individually whether they forward them or not. In contrast, for ACR only delegated agents retrieve content and deliver it to requesters. To compare the message overhead of all three schemes, we define the message overhead of Interest and Data messages as follows

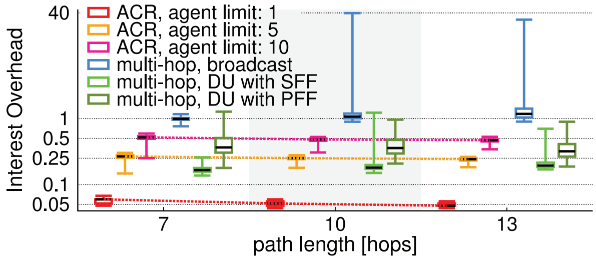
$$Overhead = \left(\frac{\sum_{i=1}^{N} m_i}{N}\right)\left(\frac{1}{S}\right), \quad (1)$$

where N is the number of nodes in the network, m_i is the number of messages sent by node i and S is the content size (number of segments). The left component in Eq. 1 denotes the average number of messages transmitted by a node. The average number is normalized by the number of segments (right component) to relate it to the number of required messages (segments). We have evaluated the message overhead separately for content sources, requesters and mobile forwarder nodes (agents). Every configuration has been evaluated in 100 different runs and the boxplots show the message overhead of all simulation runs. Because multi-hop content retrieval does not always complete for intermediate node densities within the simulation time of six hours, we only show results for high node densities in the rest of this paper.

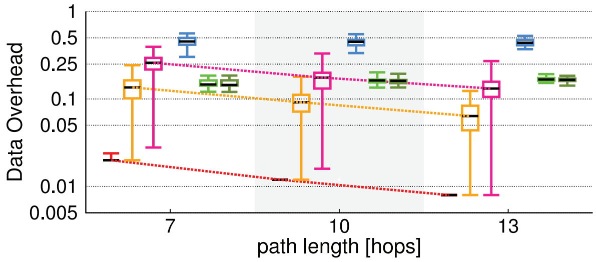
Fig. 13a shows the Interest overhead (y-axis) for different path lengths (x-axis). ACR with only one delegated agent results in the lowest Interest overhead because only one agent needs to send Interests to probe for the content while for multi-hop communication, Interests are forwarded by multiple nodes. With an increasing number of delegated agents, the number of transmitted Interests increases accordingly because agents probe for content independently. However, even for 10 delegated agents, mobile nodes send on average 50% fewer Interests than with multi-hop broadcast.

Multi-hop DU results in significantly fewer Interest transmissions than multi-hop broadcast because Interests are only transmitted on established paths and not flooded. Similarly, DU with SFF results in fewer Interest transmissions than DU with PFF because Interests are only forwarded over a single path and not multiple paths. Up to four agent delegations, ACR results in fewer Interest transmissions than multi-hop DU with SFF, but ACR requires more Interests than multi-hop DU with SFF for five agent delegations or more.

While Interest messages are rather small (around 50 bytes), Data messages have a bigger impact on network traffic because they are significantly larger (around 4500 bytes). Fig. 13b illustrates the Data overhead (y-axis) for different path lengths (x-axis). For ACR with one delegated agent, the Data overhead is negligible because only one agent needs to deliver the content to the requester. The Data overhead of multi-hop broadcast is high (mobile nodes transmit on average 50% of all Data messages), but the Data overhead can be reduced by a factor of three when using multi-hop DU (independent of the strategy) instead of multi-hop broadcast.



(a) Interest Overhead



(b) Data Overhead

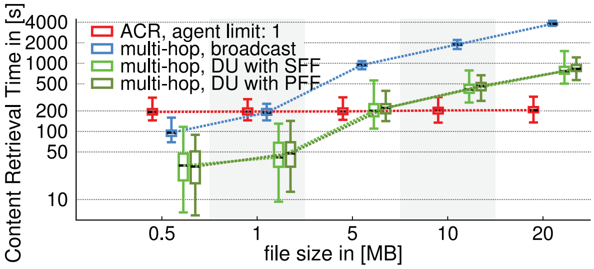
Fig. 13. Interest and Data Overhead of Mobile Nodes for a High Node Density and Varying Path Lengths.

The Data overhead of ACR with 5 and 10 delegated agents is rather high in this scenario due to the high node density. Because agents can be delegated quickly after each other, they arrive at the content source approximately at the same time. Broadcast requests from some agents may, therefore, be satisfied by the content source or other agents, which have requested the content already. This illustrates the importance for efficient agent delegation to a minimum number of agents depending on environmental conditions and application requirements. In fact, there is a tradeoff between content retrieval time, i.e., how fast content can be retrieved in an arbitrary environment, and redundant message transmissions. This tradeoff is inherent to any existing DTN routing approach [24] such as Epidemic Routing [25] or Spray-and-Wait [26]. However, even for five delegated agents, the Data overhead of ACR is similar or lower (for an increasing path length) than for DU with SFF. The Data overhead decreases with ACR for an increasing path length since more nodes are required to maintain a high node density, i.e., only delegated agents transmit messages, while the Data overhead stays rather constant (or increases slightly due to retransmissions) with multi-hop DU.

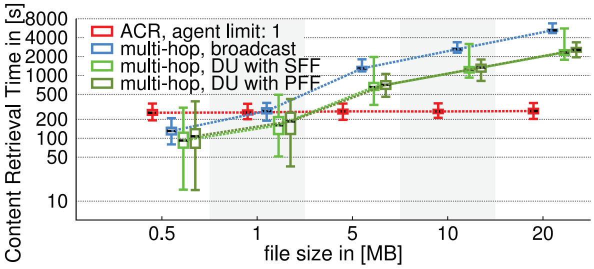
5.5. Agent-based vs. multi-hop content retrieval for multiple content sources and varying file sizes

Contact times between nodes decrease for faster node velocities. In this section, we evaluate the vehicular scenario (for a high node density) with three redundant content sources and node velocities of 10.0–14.0 m/s. Due to short contact times to content sources, we evaluate ACR with one-hop DU requests (in contrast to ACR with one-hop broadcast requests in previous subsections) and compare it to multi-hop communication. If an agent cannot com-

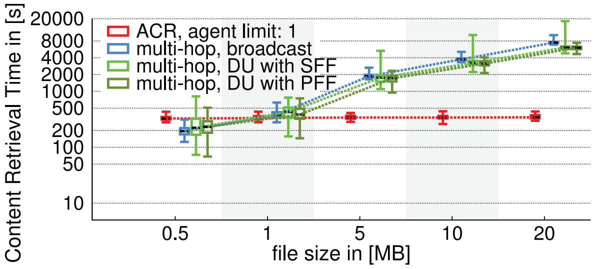
Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006



(a) Circle Radius of 250m



(b) Circle Radius of 375m



(c) Circle Radius of 500m

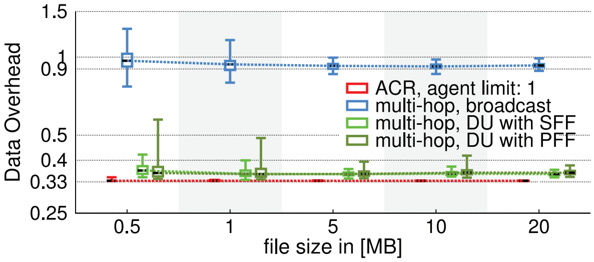
Fig. 14. Content Retrieval Times of a Requester for Different File Sizes and Circle Radii.

plete the content retrieval from one content source, it can resume it from the next content source.

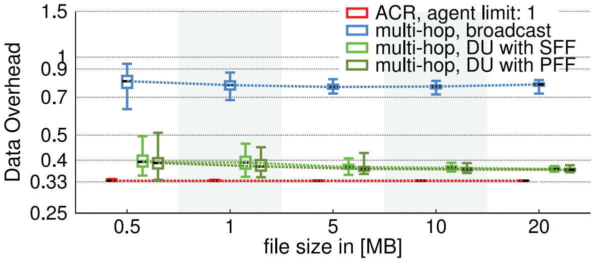
5.5.1. Content retrieval times

Fig. 14a shows content retrieval times (y-axis) for different file sizes (x-axis) on a circle with radius 250 m, i.e., at least four hops from the requester to the next content source.

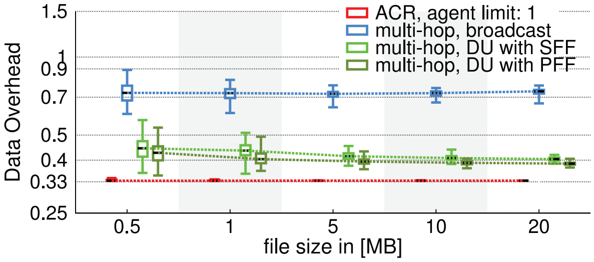
The content retrieval times with ACR are rather constant independent of the file sizes as long as they can be exchanged during the short contact time between requester and agent. Although contact times between nodes are short, i.e., a requester sees the same next hop for only 9–13 seconds (and the complete path is valid for a much shorter time), multi-hop DU performs significantly better than multi-hop broadcast. ACR is faster than multi-hop DU for larger files, i.e., 5MB and more, while multi-hop DU results in shorter content retrieval times for smaller files (1MB and less). Evaluations with slightly larger circle radii, e.g., 375 m (five hops) in Fig. 14b, look similar, but multi-hop DU requires 2–4 times longer for five hops compared to four hops while multi-hop broadcast requires only 38% more time for five hops compared to four hops. For large circle radii, e.g., 500 m (seven hops) in Fig. 14c, multi-hop DU performs similar to multi-hop broadcast because paths expire quickly, i.e., only 8–10 messages are transmitted on average before a path expires. Consequently, DU with PFF performs slightly better than DU with SFF (lower maximum values due to path redundancy). However, in such scenarios, ACR performs generally better than multi-hop communication (except for very small files below 1MB).



(a) Circle Radius of 250m



(b) Circle Radius of 375m



(c) Circle Radius of 500m

Fig. 15. Data Overhead of Content Sources for Different File Sizes and Circle Radii.

5.5.2. Transmitted messages

Fig. 15a shows the average Data overhead of all three content sources for a circle radius of 250 m.

While ACR results in (almost) perfect efficiency, i.e., a content source has a Data overhead of only 33%, multi-hop broadcast communication is very inefficient because every content source sends more than 90% of all Data messages. With multi-hop DU almost perfect efficiency can be obtained, i.e., only 2–3% more Data messages are transmitted by content sources compared to ACR. Consequently, the number of Data messages forwarded by mobile nodes in the network is also significantly lower with multi-hop DU compared to multi-hop broadcast. Figs. 15b and 15 c show the same evaluations for larger circle radii of 375 m and 500 m. Compared to a circle radius of 250 m, broadcast Data transmissions by content sources decrease by 17% (375 m) and by 24% (500 m) because of longer path lengths, i.e., the probability that some paths break and not all Interests are forwarded to all content sources increases. However, for DU with SFF and PFF the situation is different. Path breaks result in more Interest transmissions via broadcast, which address all content sources (since broadcast is the fallback strategy). For DU with SFF and PFF, Data transmissions by content sources increase by 6% for a circle radius of 375 m instead of 250 m and even by 17% (SFF) or 12% (PFF) for a circle radius of 500 m instead of 250 m. For long paths, PFF requires slightly fewer Data transmissions than SFF due to a higher path redundancy (fewer fallbacks to broadcast). Furthermore, although content retrieval with multi-hop DU requires approximately the same time than multi-hop broadcast (cf. Fig. 14c), Fig. 15c shows that multi-hop DU still results in fewer Data transmissions than multi-

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

hop broadcast. This indicates that multi-hop DU experiences many timeout periods, where Interests are not forwarded due to path breaks and can not be retransmitted until they expire (the default Interest lifetime is four seconds). Thus, to quickly react to path breaks and increase throughput of multi-hop DU, adaptive Interest lifetimes [40] based on measured round-trip times (RTTs) may be beneficial.

For ACR, the path length has no significant impact on the number of Data transmissions by content sources but the contact time to content sources does. Therefore, we have also evaluated the overhead of ACR with one-hop DU when retrieving large files from multiple content sources, i.e., resume operations. However, the overhead is negligible, e.g., for a download of 60MB (15,000 segments) at a mobile speed of 14 m/s from three content sources only 5–8 more segments need to be transmitted by content sources (because the agent is out of range for reception) and agents send 10–15 more Interest messages via unicast (because disruptions can not be detected instantaneously).

6. Lessons learned

6.1. Push vs. pull notification

After an agent has completed content retrieval, it needs to notify the requester. We have observed that pull notifications are more efficient for multiple delegated agents since one message can address multiple agents at the same time. In addition, pull notifications are rather small, i.e., they request additional information only when a suitable agent is in range, such that they result in less network traffic than push notifications. On the downside, a requester needs to transmit pull notifications without knowing whether an agent has already retrieved the content, while push notifications are only transmitted by agents after finding the content. Thus, for pull notifications, a requester needs to estimate when to start requesting notifications (or just start after delegating the first agent).

6.2. Agent selection and delegation

When delegating agents, there is a tradeoff between content retrieval time and message overhead. Thus, it is important to keep the number of delegated agents at the lowest possible level which still enables reasonably fast content retrieval. In our current implementation, all agents are delegated in short time intervals to quickly retrieve content even in the presence of many slow nodes. An alternative (more conservative) strategy could be to delegate content retrieval to more agents only after a much larger time interval, e.g., based on estimations when an agent might return (past experiences).

Furthermore, the quality of agent selection can be improved with additional information. Currently, we select agents in a requester's neighborhood randomly. Other options for more efficient agent selection may be based on social criteria, e.g., social interactions [41] or reputation [28], past experiences, e.g., overheard content [38] or past GPS traces [42], or based on hybrid approaches [43,44], where potential agents are discovered from a central server (when connected to the Internet) for later usage when disconnected from the Internet.

6.3. Impact of path length

In dense environments, multi-hop DU results in faster content retrieval times than ACR. Yet, if only a few agents are delegated, ACR has a lower message overhead because only selected agents need to transmit messages while for multi-hop communication every node on the path forwards messages. For ACR, the message overhead decreases with increasing path length while it stays constant (or slightly increases due to retransmissions) for multi-hop DU.

In general, multi-hop DU performs better than broadcast communication for long path lengths because no broadcast delays (for duplicate suppression) are required during unicast communication. Hence, breaking of symmetric Interest-Data forwarding paths is no issue because Data can be returned quickly (within milliseconds), i..e, the topology has not changed much.

6.4. Impact of node velocity

DU with SFF is more efficient than DU with PFF for pedestrian mobility with respect to content retrieval times and message transmissions. If nodes move with vehicular velocities, neighboring nodes may see each other only for a short time. Thus, for long path lengths, multi-hop DU results in similar content retrieval times than multi-hop broadcast due to frequent path breaks (timeouts and fallbacks to broadcast). In particular, DU with PFF performs slightly better than DU with SFF for long paths due to higher path redundancy.

Yet, ACR performs generally better than DU with either forwarding strategy in high speed scenarios. To retrieve content in case of fast node velocities, it is crucial to detect available content sources quickly. In our evaluations, agents periodically probed for content (Interest probing) at a fixed rate of one Interest per second. However, Interest probing may be adapted based on node velocity, e.g., fewer Interests at lower speed, location or past experience to minimize the Interest overhead.

Furthermore, to increase throughput during short contact times, ACR can be combined with one-hop DU, which addresses Interests (after an initial broadcast) at a higher rate to the same content source until it becomes unavailable. We have seen that the overhead of one-hop DU for resume operations from multiple content sources is negligible.

6.5. Combination of ACR and DU

ACR and multi-hop DU perform differently under similar network conditions. In general, multi-hop DU performs better for content retrieval in dense environments and for small content objects while ACR performs better in sparse environments, for fast node velocities and for large content objects. Therefore, ACR and multi-hop DU can complement each other perfectly. In dense or sparse environments, the combination is straightforward, i.e., a requester can try to retrieve content via multi-hop and delegate it to an agent only if multi-hop content retrieval is not successful. However, in intermediate (neither dense nor sparse) environments, the combination is more complex and requires further investigations. In particular, it needs to be explored how quickly a requester should delegate content retrieval to an agent after a disruption. If it is delegated too early and connectivity would be re-gained quickly, there may be redundant message transmissions. However, if a requester waits a long time before delegating content retrieval to an agent, the content retrieval time increases accordingly.

6.6. Security

Delegating content retrieval to other nodes introduces various attack options, which have not been analyzed in this work. For example, an agent should not retrieve malicious or illegal content for other users. To mitigate this threat, trust and reputation models may be established. Then, agents and requesters could exchange their identities and sign messages during agent delegations.

Furthermore, the retrieved content should not be too large and fill the agent's complete memory. Therefore, requesters and agents can negotiate a maximum content size during agent delegation. If the content is larger than an agent can handle, a requester needs to delegate content retrieval to multiple agents, which request different parts of the content. An approach to do this may be based on RC-NDN [45], where content sources use Raptor codes to create Data encodings. Without requiring coordination, agents can retrieve a certain number of Data encodings and deliver it to the requester, where original content can be recovered after decoding.

7. Conclusions

We have described agent-based content retrieval (ACR) and showed that delay-tolerance in information-centric networks (ICN) can be supported without modifications to ICN message processing. This enables seamless operation in well-connected and disruptive networks. Furthermore, we have shown that mobile ICN communication does not require all messages to be transmitted via broadcast. Dynamic Unicast (DU) has resulted in faster content retrieval times than broadcast for slow and high node velocities (up to a certain path length). Symmetric Interest-Data forwarding paths have not been identified as limitation because Data messages are returned within milliseconds, i..e, the topology has not changed much.

We have seen that node mobility is not necessarily a disadvantage for wireless communication and ICN provides the means to exploit it. While multi-hop communication is faster with high node densities, ACR is superior in low and intermediate node densities where multi-hop communication does not work or results in frequent disruptions. Furthermore, ACR is beneficial for large file sizes and works well even under high mobility, where it can be combined with one-hop DU. Although, in our scenarios, agents had to return to the requester to deliver content, agent delegation and content delivery can also be at different locations as long as both locations can communicate and coordinate with each other.

With our approach, multi-hop DU and ACR can be combined. A requester could initially try to retrieve content via multi-hop communication and only delegate retrieval to agents if nothing has been found. Because all messages are stored in the same ICN message format, requesters could also retrieve (via multiple hops) content from agents, which were delegated by other requesters. However, concrete mechanisms to combine multi-hop DU with ACR under dynamic (time varying) network condition are still a subject for more investigations.

References

[1] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, K. Leung, A survey on the IETF protocol suite for the internet of things: standards, challenges, and opportunities, IEEE Wireless Commun. Mag. 20 (6) (2013) 91–98.
[2] E. Baccelli, C. Mehlis, O. Hahm, T.C. Schmidt, W. Wählisch, Information centric networking in the IoT: experiments with NDN in the wild, in: 1st ACM Conference on Information-Centric Networking (ICN), Paris, France, 2014, pp. 77–86.
[3] M. Amadeo, C. Campolo, A. Molinaro, Multi-source data retrieval in IoT via named data networking, in: 1st ACM Conference on Information-Centric Networking (ICN), Paris, France, 2014, pp. 67–76.
[4] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, A. Mauthe, A survey of mobility in information-centric networks, ACM Commun. Rev. 56 (12) (2013) 90–98.
[5] M. Meisel, V. Pappas, L. Zhang, Ad hoc networking via named data, in: 5th ACM MobiArch, Chicago, USA, 2010, pp. 3–8.
[6] J. Wang, R. Wakikawa, L. Zhang, DMND: collecting data from mobiles using named data, in: Proceedings of 2nd IEEE Vehicular Networking Conference, 2010.
[7] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, L. Zhang, Rapid traffic information dissemination using named data, in: Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications, Hilton Head Island, SC, USA, 2012, pp. 7–12.
[8] M. Amadeo, A. Molinaro, G. Ruggeri, E-CHANET: routing, forwarding and transport in information-centric multihop wireless networks, Comput. Commun. 36 (2013) 792–803.
[9] Y. Yu, C. Tandiono, X. Li, Y. Lu, M.Y. Sanadidi, M. Gerla, ICAN: information-centric context-aware ad-hoc network, in: International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, 2014, pp. 578–582.
[10] G. Grassi, D. Pesavento, G. Pau, L. Zhang, S. Fdida, Navigo: interest forwarding by geolocations in vehicular named data networking, in: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Boston, MA, USA, 2015.
[11] M.C. Gonzalez, C.A. Hidalgo, L. Barabasi, Understanding individual human mobility patterns, Nature 453 (5) (2008) 779–782.
[12] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, R.L. Braynard, Network named content, in: 5th ACM CoNEXT, Rome, Italy, 2009, pp. 1–12.
[13] The CCNx Project, URL http://www.ccnx.org/, last accessed: April 2016.
[14] NS-3 Direct Code Execution, URL http://www.nsnam.org/overview/projects/direct-code-execution/, last accessed: April 2016.
[15] Ubelix, Linux Cluster of University of Bern, URL http://www.ubelix.unibe.ch, last accessed: April 2016.
[16] Named Data Networking (NDN) Project. URL http://named-data.net/, last accessed: April 2016.
[17] NDNx: The NDN Platform, URL http://named-data.net/codebase/platform/, last accessed: April 2016.
[18] M. Mosko, I. Solis, CCNx Semantics, Internet Draft, January 2016. URL https://tools.ietf.org/html/draft-irtf-icnrg-ccnxsemantics-01, last accessed: April 2016.
[19] CCNx Interest Message, URL https://www.ccnx.org/releases/ccnx-0.8.2/doc/technical/InterestMessage.html, last accessed: April 2016.
[20] NDNx Interest Packet, URL http://named-data.net/doc/NDN-TLV/current/interest.html, last accessed: April 2016.
[21] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, L. Zhang, VANET via named data networking, in: IEEE Infocom workshop on Name Oriented Mobility (NOM), Toronto, ON, Canada, 2014, pp. 410–415.
[22] C. Anastasiades, T. Braun, Dynamic transmission modes to support opportunistic information-centric networks, in: International Conference on Networked Systems (NetSys), Cottbus, Germany, 2015, pp. 1–5.
[23] S. Jain, K. Fall, R. Patra, Routing in a delay tolerant network, in: ACM SIGCOMM, Portland, OR, USA, 2004, pp. 145–158.
[24] M. Liu, Y. Yang, Z. Qin, A survey of routing protocols and simulations in delay–tolerant networks, Wireless Algo., Syst. Appl. 6843 (2011) 243–253.
[25] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Technical Report, Duke University, 2000.
[26] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: ACM SIGCOMM workshop on Delay Tolerant Networks (WDTN), Philadelphia, PA, USA, 2005, pp. 252–259.
[27] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, ACM SIGMOBILE Mobile Comput. Commun. Rev. 7 (3) (2003) 19–20.
[28] Y. Zhu, B. Xu, X. Shi, Y. Wang, A Survey of social-based routing in delay-tolerant networks: positive and negative social effects, IEEE Commun. Surveys Tuts. 15 (1) (2013) 387–401.
[29] N. Orlinski, N. Filer, Neighbor discovery in opportunistic networks, Ad Hoc Netw. 25 (2015) 383–392.
[30] T. Hossmann, T. Spyropoulos, F. Legendre, Know thy neighbor: towards optimal mapping of contacts to social graphs for DTN routing, in: IEEE INFOCOM, San Diego, CA, USA, 2010, pp. 866–874.
[31] R.N. dos Santos, B. Ertl, C. Barakat, T. Spyropoulos, T. Turletti, CEDO: content–centric dissemination algorithm for delay-tolerant networks, in: Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems, Barcelona, Spain, 2013, pp. 377–386.
[32] E. Monticelli, B.M. Schubert, M. Arumaithurai, X. Fu, K.K. Ramakrishnan, An information centric approach for communications in disaster situations, in: 20th International workshop on Local & Metropolitan Area Networks (LANMAN), Reno, NV, 2014, pp. 1–6.
[33] M. Pitkänen, T. Kärkkäinen, J. Ott, Mobility and service discovery in opportunistic networks, in: IEEE PerMoby, Lugano, Switzerland, 2012.
[34] C. Anastasiades, W. El Maudni El Alami, T. Braun, Agent-based content retrieval for opportunistic content-centric networks, in: WWIC, LNCS 8458, Paris, France, 2014, pp. 175–188.
[35] CCNx Basic Name Conventions, URL https://www.ccnx.org/releases/ccnx-0.8.2/doc/technical/NameConventions.html, last accessed: April 2016.
[36] C. Anastasiades, A. Sittampalam, T. Braun, Content discovery in wireless information-centric networks, in: 40th IEEE Conference on Local Computer Networks (LCN), Clearwater Beach, FL, USA, 2015, pp. 237–245.
[37] CCNx Repository Protocol, URL https://www.ccnx.org/releases/ccnx-0.8.2/doc/technical/RepoProtocol.html, last accessed: April 2016.
[38] C. Anastasiades, T. Braun, Towards information-centric wireless multihop communication, in: 13th International Conference on Wired & Wireless Internet Communications (WWIC), LNCS 9071, Malaga, Spain, 2015, pp. 367–380.
[39] C. Anastasiades, T. Schmid, J. Weber, T. Braun, Opportunistic content-centric data transmission during short network contacts, in: IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, 2014, pp. 2516–2521.
[40] C. Anastasiades, L. von Rotz, T. Braun, Adaptive interest lifetimes for information-centric wireless multi-hop communication, in: 8th IFIP Wireless and Mobile Networking Conference (WMNC), Munich, Germany, 2015, pp. 40–47.
[41] A.G. Miklas, K.K. Gollu, K. Chan, S. Saroiu, K.P. Gummadi, E. de Lara, Exploiting social interactions in mobile systems, in: 9th International Conference on Ubiquitous Computing (UbiComp), Innsbruck, Austria, 2007, pp. 409–428.
[42] A. Hess, J. Ott, Extrapolating sparse large-scale GPS traces for contact evaluation, in: 5th ACM workshop on HotPlanet, Hong Kong, 2013, pp. 39–44.

[43] T. Hossmann, D. Schatzmann, P. Carta, F. Legendre, Twitter in disaster mode: smart probing for opportunistic peers, in: 3rd ACM International workshop on Mobile Opportunistic Networks (MobiOpp), Zurich, Switzerland, 2012, pp. 93–94.

[44] A. Socievole, F.D. Rango, S. Marano, Face-to-face with facebook friends: using online friendlists for routing in opportunistic networks, in: 24th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications: Mobile and Wireless Networks (PIMRC), London, UK, 2013, pp. 2989–2994.

[45] C. Anastasiades, N. Thomos, A. Striffeler, T. Braun, RC-NDN: raptor codes enabled named data networking, in: IEEE International Conference on Communications (ICC), London, UK, 2015, pp. 3026–3032.

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006

Carlos Anastasiades received his M.Sc. degree in Electrical Engineering and Information Technology from ETH Zurich, Switzerland in 2009. Since then, he has been a Ph.D. student and research assistant at University of Bern, Switzerland. His research interests include routing and caching in wireless information-centric networks, mobile ad-hoc networks and delay-tolerant networks.

Tobias Schmid received his M.Sc. degree in Computer Science from University of Bern, Switzerland in 2015. Since 2012, he has been investigating opportunistic information-centric communication in wireless networks. His research interests lie in the area of mobile and delay-tolerant information-centric networks.

Jürg Weber received his M.Sc. degree in Computer Science from University of Bern, Switzerland in 2015. During his B.Sc. and M.Sc. studies, he has been investigating efficient forwarding strategies in information-centric wireless mesh networks. His research interests lie in the area of routing and efficient data transmission in mobile networks.

Torsten Braun got his Ph.D. degree from University of Karlsruhe (Germany) in 1993. From 1994 to 1995 he has been a guest scientist at INRIA Sophia-Antipolis (France). From 1995 to 1997 he has been working at the IBM European Networking Centre Heidelberg (Germany) as a project leader and senior consultant. He has been a full professor of Computer Science at the University of Bern (Switzerland) and head of the research group “Communication and Distributed Systems” since 1998. He has been member of the SWITCH (Swiss education and research network) board of trustees since 2001. Since 2011, he has been vice president of the SWITCH foundation.

Please cite this article as: C. Anastasiades et al., Information-centric content retrieval for delay-tolerant networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.006