ARTICLE IN PRESS
JID: COMPNW [m5G;April 11, 2016;19:15]

Computer Networks 000 (2016) 1–12

Contents lists available at ScienceDirect

Computer Networks

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

Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks

Carlos Anastasiades*, Jürg Weber, Torsten Braun

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

ARTICLE INFO

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

Keywords:
Information-centric
Wireless routing
Mobile
Multi-hop
Broadcast
Unicast
ICN
CCN

ABSTRACT

Information-centric networking (ICN) offers new perspectives on mobile ad-hoc communication because routing is based on names but not on endpoint identifiers. Since every content object has a unique name and is signed, authentic content can be stored and cached by any node. If connectivity to a content source breaks, it is not necessarily required to build a new path to the same source but content can also be retrieved from a closer node that provides the same content copy. For example, in case of collisions, retransmissions do not need to be performed over the entire path but due to caching only over the link where the collision occurred. Furthermore, multiple requests can be aggregated to improve scalability of wireless multi-hop communication. In this work, we base our investigations on Content-Centric Networking (CCN), which is a popular ICN architecture. While related works in wireless CCN communication are based on broadcast communication exclusively, we show that this is not needed for efficient mobile ad-hoc communication. With *Dynamic Unicast* requesters can build unicast paths to content sources after they have been identified via broadcast. We have implemented Dynamic Unicast in CCNx, which provides a reference implementation of the CCN concepts, and performed extensive evaluations in diverse mobile scenarios using NS3-DCE, the direct code execution framework for the NS3 network simulator. Our evaluations show that Dynamic Unicast can result in more efficient communication than broadcast communication, but still supports all CCN advantages such as caching, scalability and implicit content discovery.

1. Introduction

Mobile ad-hoc networks (MANETs) are composed of mobile nodes that create ad-hoc connections among themselves to enable communication in areas without communication infrastructures. MANETs have been an active research area for more than two decades and diverse host-based proactive and reactive routing protocols have been proposed [1–4]. Host-based routing protocols explore routing paths to destinations by forwarding messages based on endpoint identifiers. However, in case of high mobility, routing paths may break such that communication is disrupted until a path to a new content source can be found. In host-based communication, content retrieval cannot be seamlessly resumed from an alternative content source since handover mechanisms are required to discover endpoints that can serve the same content. Furthermore, host-based communication may overload a network or a server if popular content is requested by many users since the traffic scales with the number of requests from different users (popularity).

Recently, information-centric networking (ICN) [5] has been proposed to address shortcomings of host-based communication such as scalability, consumer mobility and security. In ICN, requests are routed based on names towards content sources. Since, every content object has a unique name and is signed by the publisher, authentic content can be cached at and retrieved from any node. We base our work on the Content-Centric Networking (CCN) [6] architecture, where users can broadcast Interests to receive Data on the reverse path independent of the content location. Then, if connectivity to a content source breaks, e.g., because the distance becomes too long, requests can be satisfied by nearby nodes that store the same content. In case of collisions or disruptions due to mobility, content does not have to be retransmitted over the entire path but can be retrieved from the closest intermediate cache. Furthermore, it is possible to identify identical user requests to suppress redundant message transmissions and save energy. Hence, assuming that energy is a critical resource on mobile devices, fewer message transmissions may prolong a device's lifetime.

---

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

http://dx.doi.org/10.1016/j.comnet.2016.03.009
1389-1286/© 2016 Elsevier B.V. All rights reserved.

It has been shown that wireless ICN can outperform existing host-based communication protocols such as Mobile IP [7] and AODV [8,9] in terms of throughput and message efficiency in highly mobile networks. Most wireless ICN routing protocols use broadcast communication exclusively [8–12] to enable content retrieval from any content source (source flexibility). However, if all messages are transmitted via broadcast and received by all nodes, the message processing and energy overhead of every node would increase (due to more message transmissions and fewer sleep cycles). Furthermore, if nodes do not have sufficiently large caches, there is no benefit in receiving and caching overheard Data messages because they are quickly replaced by new content.

In this paper, we evaluate *Dynamic Unicast*, which does not maintain source flexibility via broadcast for every single Interest message. Instead, requesters perform flooding only to find a content source and forward future requests directly to the same source using unicast. If the content source becomes unavailable, requesters can quickly revert to broadcast to find another content source. The contributions of this paper are as follows:

- We describe Dynamic Unicast, an information-centric routing protocol for mobile ad-hoc networks. Dynamic Unicast is based on implicit content discovery via broadcast and dynamically created unicast links for efficient content retrieval. In contrast to related work, Dynamic Unicast does not require location information or any modifications to ICN messages and can, therefore, be easily integrated with (partially) wired networks.
- We describe two forwarding strategies for Dynamic Unicast and compare them in different scenarios to broadcast communication. Our evaluations show that Dynamic Unicast is more efficient than broadcast communication, but it still supports caching, improved scalability with multiple requesters and implicit content discovery.
- We describe and evaluate a Content Request Tracker, which is a data structure that keeps track of incoming unicast requests, to replace multiple unicast transmissions with one broadcast transmission (for improved communication efficiency).

The remainder of this paper is organized as follows. In Section 2 we describe CCN and related work on wireless routing protocols. Section 3 describes *Dynamic Unicast* and forwarding strategies for information-centric wireless multi-hop communication. In Section 4, a *Content Request Tracker* is introduced as optional extension to optimize message transmissions in case of multiple requesters. Evaluation results are shown 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) [6] is a popular ICN architecture that has been implemented in the CCNx Project [13]. In CCN, content is organized in segments, which are included in Data messages signed by the content publisher. To retrieve a content object, requesters need to sequentially transmit Interest messages for every Data message of a content object. The final segment is marked in the header of a Data message. Content names follow a hierarchical structure

$$\underbrace{/c_0/\cdots/c_n/fname/version/}_{\text{Content Prefix for Dynamic Unicast}}s_n$$

which is composed of arbitrary name components $c_0, \ldots, c_n$, a file name *fname*, a *version* number that labels different versions of the same content, and a segment number s_n that marks each segment individually. To ensure globally unique names and support (longest-prefix) routing, content names may be aggregated by publisher specific prefixes.

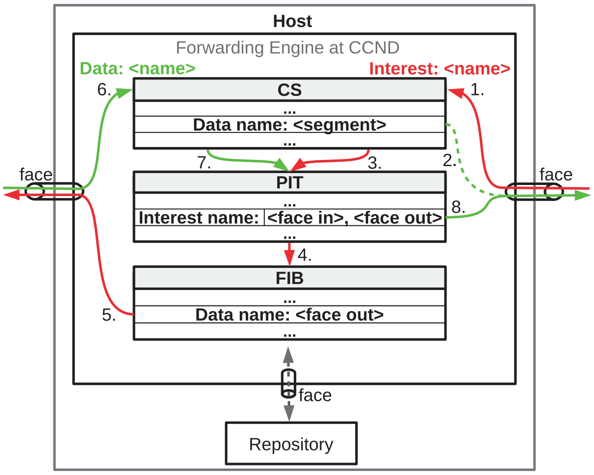


Fig. 1. CCN message processing.

In CCNx, message processing at each host is performed by the CCN daemon (CCND) as illustrated in Fig. 1. Links to local applications or other hosts are called faces.

The Content Store (CS) is a cache that holds received and forwarded Data temporarily. When receiving an Interest, it is first checked whether the requested Data message can be found in the CS (step 1). If this is the case, the Data message can be returned (step 2) without further propagating the Interest. If Data cannot be found in the cache, the Pending Interest Table (PIT) is consulted (step 3). The Pending Interest Table (PIT) records from which faces Interests have been received and to which faces they have been forwarded. A PIT entry is removed if matching Data comes back or if the Interest expires (based on the Interest lifetime defined in the Interest header). Existing PIT entries prevent forwarding of similar Interests (Interest aggregation). If there is no PIT entry (step 4), the Forwarding Information Base (FIB) defines over which faces Interests can be forwarded towards a content source (step 5). If there are no matching FIB entries, Interests are dropped. After receiving a Data message in return to an Interest, it is stored in the CS (step 6). The freshnessSeconds field in the Data header determines how long a cached Data message remains valid until it expires. Based on recorded PIT information (step 7), Data messages can be forwarded on the reverse path towards requesters (step 8). To enable duplicate suppression, broadcast Data transmissions are scheduled with a broadcast delay. Then, if a node overhears the transmission of the same Data message from another node, it can cancel a scheduled Data transmission.

Content can be persistently stored and provided to others by repositories. Repositories are implemented as local applications, i.e., they receive and transmit messages via internal face to the CCND.

2.2. Mobile and wireless ad-hoc routing

Existing MANET routing protocols have functional components for path discovery, data dissemination and route maintenance. *Path discovery* is required to learn end-to-end paths between source and destination nodes, which can later be used for data dissemination. In proactive protocols such as Optimized Link State Routing (OLSR) [14], nodes select multipoint relays (MPRs) among two-hop neighbors and exchange topology control messages to create routing tables to all available destinations. In reactive protocols, end-to-end paths are created on-demand by flooding route requests and establishing paths via route replies transmitted on the

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009

reverse path via unicast. In Dynamic Source Routing (DSR) [15], the complete forwarding node sequence is included in route replies such that sources learn complete paths to destinations. In Ad-hoc On-Demand Distance Vector Routing (AODV) [3] and AODVv2 [4] forwarding nodes establish soft states to neighbors to create hop-by-hop forwarding paths between sources and destinations.

After paths have been discovered, *data dissemination* is performed via unicast on the discovered paths. In case of mobility, *route maintenance* mechanisms are required to detect path breaks and keep routing information up-to-date. If proactive protocols detect path breaks, they need to re-calculate network structures (e.g., MPRs), and routing tables. Hence, they target rather static networks. If reactive protocols detect path breaks, they remove invalid routing information and perform a new path discovery to the same destination. However, if the previous destination is unreachable or very far away, complex handover mechanisms are required to find alternative content sources nearby. In contrast, CCN requesters can seamlessly find new content sources since requests are routed based on content names.

Routing in Wireless Sensor Networks (WSNs, see [16] for a survey) has similarities to CCN because routing is data-centric, i.e., data is requested based on names while node IDs of responding nodes are not important. A popular data-centric WSN routing protocol is Directed Diffusion (DD) [17]. In DD, a sink broadcasts Interests for names to find sensors with matching Data. The Interests establish soft states in forwarding nodes such that Data can travel at a low rate via unicast on the reverse path. In contrast to CCN, these soft states are not deleted after receiving a data packet but maintained to receive data at a specific rate until they expire. A sink can re-inforce soft states positively (higher data rate) or negatively (lower data rate).

Despite similarities to CCN, data-centric WSN routing protocols cannot be used as general purpose MANET routing protocols. Since sensor networks are built for a specific purpose, e.g., monitoring the temperature, sensors can understand data values and aggregate them. In CCN, data aggregation is not possible by intermediate nodes because they (i) may not understand the data and (ii) would need to add new signatures when modifying data. Furthermore, WSNs target typically rather static scenarios with a sink node that collects rather redundant data from multiple sensors, while MANETs can be highly mobile comprising multiple requesters and diverse content sources.

2.3. Wireless ICN routing

Since ICN namespaces can be several orders of magnitude larger than current IP address spaces and mobile topologies may change dynamically, only reactive wireless ICN routing protocols have been proposed for MANETs. In fact, analytical work [18] has shown that maintaining routing information in mobile networks is costly such that flooding may be beneficial in case of high host churn. Since routing is based on names, path discovery (also called content discovery) can be performed implicitly while flooding requests for content.

Listen-First-Broadcast-Later (LFBL) [8], E-CHANET [9] and Content-Centric Vehicular Networking (CCVN) [19] extend ICN messages with additional fields including endpoint IDs (requester and content source), hop count and hop distance between endpoints. If requesters do not know a content source, they flood Interest messages until they receive a Data reply from a content source (with all fields filled). Then, the next Interests are addressed to the same content source by specifying content source ID and hop distance. Yet, LFBL, E-CHANET and CCVN transmit all messages via broadcast to benefit from increased content density (caching) and enable path maintenance (hop distances to content sources) by overhearing Interest and Data messages. However, if endpoint IDs are added to both Interest and Data messages, the communication is not strictly information-centric anymore. For example, handovers are required [9] to match Interests with cached Data messages when they contain different content source IDs. Similarly, Interest aggregation for multiple requesters becomes more complex because of different endpoint IDs and potentially different hop distances. In particular, hop distances in cached Data messages may be inaccurate in case of mobility.

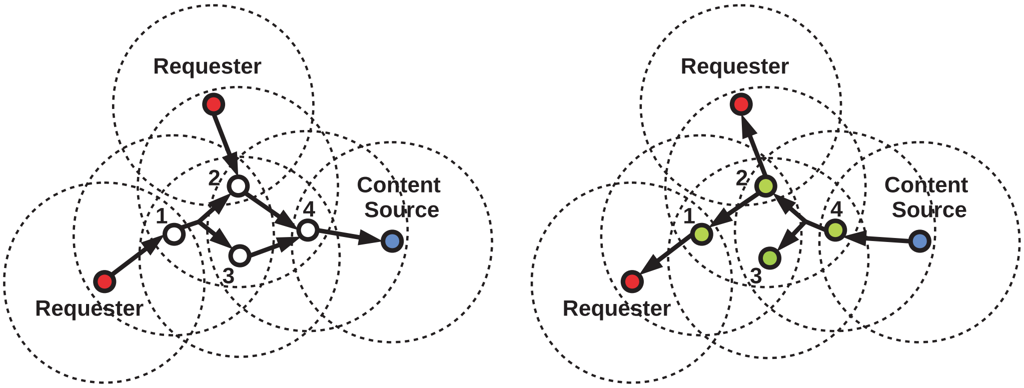
Wireless ICN communication without endpoint identifiers is beneficial in case of high mobility [7] because ICN requests can seamlessly retrieve content from any content source without topology discovery. To ensure quick dissemination of ICN messages, location information can be encoded in names [10] enabling nodes farther away to re-broadcast packets earlier to make more progress. However, such a strategy would require forwarding strategies to understand name semantics.

To avoid complex forwarding strategies based on semantics, a Link Adaptation Layer (LAL) [20] has been proposed to enable broadcast support with location information (GPS coordinates) for raw 802.11 frames. By that, each node can compute the distance to a previous sender and trigger message forwarding based on expected progress and overheard transmissions. Navigo [12] is a geographic ICN routing protocol based on LAL that binds content names to a producer's geographic area (squares on a digital map). Then, during an initial flooding, content sources can attach their geographic area in returned Data messages such that future requests can be directed towards the same area. While Navigo enables efficient routing over large distances (although requiring localization mechanisms), it also results in an increased content density due to broadcast Data transmissions (caching). Hence, to avoid duplicate Data transmissions for subsequent broadcast requests, large broadcast delays are required [21]. However, large broadcast delays reduce the achievable throughput for wireless communication. In static networks, it may be possible to successively reduce broadcast delays by identifying preferred forwarders [22], but this may not work efficiently in highly mobile networks when preferred forwarders may regularly move away.

While broadcast delays are necessary during broadcast communication to enable duplicate suppression, they are not required during unicast transmissions. Therefore, it may be beneficial to avoid broadcast communication whenever unicast communication is possible. Reactive Optimistic Name-based Routing (RONR) [23], which has been designed for static sensor networks, uses broadcast to find a content source and configures unicast faces in the FIB on the reverse path. After a discovery phase, subsequent Data can be retrieved from the content source via unicast path. In static IoT scenarios, RONR can reduce the number of radio transmissions by up to 50% compared to broadcast and, thus, save energy on resource constrained devices. However, RONR has not been designed for and evaluated in mobile ad-hoc networks with changing connectivity patterns that require dynamic Interest forwarding and FIB updates.

We have seen that dynamically created unicast faces are beneficial for opportunistic one-hop communication [21] because short contacts to content sources can be better exploited due to faster content retrieval times (no broadcast delays) and fewer duplicate Data transmissions. In this paper, we extend previous work on Dynamic Unicast [21] in multiple ways. First, we extend one-hop Dynamic Unicast for multi-hop communication. This requires new Interest forwarding strategies and FIB update mechanisms to keep forwarding information accurate. Second, we implement a Content Request Tracker to perform unicast or broadcast transmissions depending on the number of concurrent requesters. Third, we implement all described mechanisms in CCNx and not only a network simulator. Forth, we evaluate the described mechanisms in diverse scenarios with different topologies, node velocities and content

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009



(a) Requesters flood Interest messages.

(b) Content source replies Data via broadcast. Nodes register unicast face after receiving Data message.

Fig. 2. Implicit content discovery.

source densities (previous work [21] has only been evaluated with one content source). The described mechanisms do not require GPS coordinates and can be used without modifications of CCN Interest and Data messages.

3. Dynamic Unicast

In this section, we describe *Dynamic Unicast*, which creates Dynamic Unicast faces to locally available content sources that are discovered via broadcast requests.

3.1. Protocol overview

Fig. 2 illustrates implicit content discovery with Dynamic Unicast. Requesters broadcast Interest messages, which are forwarded by other nodes in transmission range. If multiple nodes receive an Interest via broadcast, e.g., nodes 2 and 3 in Fig. 2a, they may forward it simultaneously resulting in a duplicate Interest transmission. If either node 2 or 3 forwards the Interest slightly before the other node, a duplicate Interest transmission may be prevented because the same Interest is already included in the PIT. Hence, similar Interests from multiple requesters can be aggregated in the PIT such that only one Interest is forwarded, e.g., at node 2 and 4 in Fig. 2a. If Interests reach a content source, Data messages return on the reverse path via broadcast as illustrated in Fig. 2b. All nodes that overhear the Data transmission can configure a unicast face to the previous hop in the FIB. Duplicate Data transmissions, e.g., at node 3 in Fig. 2b, may be prevented since broadcast Data transmissions are delayed to enable duplicate suppression (see Section 2.1).

After the Data packet has reached the requesters, a hop-by-hop unicast path to the content source has been configured in the FIBs of all intermediate nodes. Thus, content retrieval can be performed via unicast as illustrated in Fig. 3. Although transmissions are performed via unicast, Interest aggregation from different requesters and Data caching is still supported.

3.2. Enabling multi-hop communication

To discover available content sources and enable Dynamic Unicast, broadcast Interests need to be transmitted (see Section 3.1). However, if no broadcast FIB entries are configured for the name prefix, the corresponding Interests are dropped. To address this issue, we have implemented a *pass-through* mechanism, which redirects Interests without matching FIB entry to a broadcast face. In previous work [21], we enabled pass-through only for local applications but in this work we extend it for Interests from other hosts to enable multi-hop communication. To limit Interest forwardings by nodes that cannot reach a content source, users can define an upper limit of Interests that can be forwarded via pass-through. For example, a node may only allow one pass-through per content name, i.e., more Interests can only be forwarded if Data returns and a prefix is configured in the FIB or if the Interest expires and is removed from the PIT.

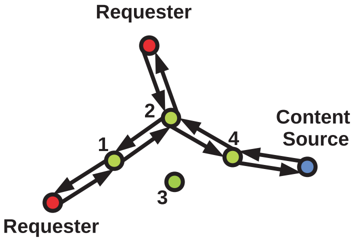


Fig. 3. Content retrieval via configured unicast faces. Interest aggregation and Data caching is still possible.

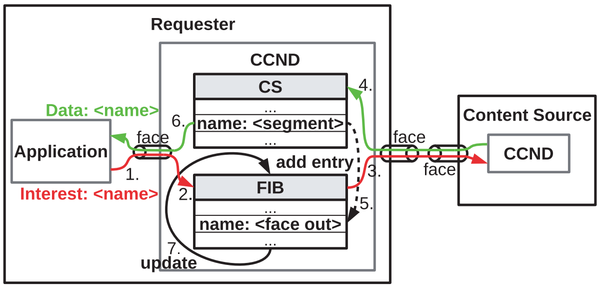


Fig. 4. Overhearing of Data messages to configure Dynamic Unicast faces in the FIB.

3.3. Dynamic prefix registration

Fig. 4 illustrates the prefix registration process at a requester. Applications initiate content retrieval by transmitting Interests via internal face to the CCND (step 1), where a FIB lookup is performed (step 2). If the FIB does not contain a unicast face to a content source, the Interest is transmitted via broadcast (step 3) towards any content source in the vicinity (pass-through). Forwarded Interests are always included in the PIT (not shown in Fig. 4). After receiving a Data message on the reverse path (step 4), dynamic FIB entries can be configured towards the content source (step 5) and the Data is forwarded further (step 6) based on PIT information.

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009

To maintain accurate forwarding information, the FIB is regularly updated (step 7, see Section 3.4). Dynamic FIB entries based on received Data messages are only configured if the corresponding Interests have been transmitted and the Data is new, i.e., not already in the cache indicating that the Data has been received previously. To configure a dynamic FIB entry, the node ID of the previous hop and the content prefix (content name without segment number) are extracted. In our implementation, we use IP addresses as node IDs, however, other node IDs such as MAC addresses or descriptive names would also be possible. These node IDs are not included in CCN messages but can be extracted from headers of packets that have transported CCN messages over the previous hop, e.g., IP or MAC packets. Hence, we do not use IP addresses for global end-to-end routing but only to identify a next hop towards a content source.

3.4. Updating the FIB

In mobile networks, connectivity to other nodes may change quickly. Thus, it is crucial to remove outdated information from the FIB as quickly as possible because Interests transmitted over broken paths increase message overhead and transmission times. We delete expired information in two cases.

First, we perform periodic update operations in the FIB to track the number of received messages over a face. If no messages have been received via a Dynamic Unicast face for some time, e.g., if the neighbor is not in range anymore, the corresponding face is automatically deleted. If faces are deleted, the corresponding FIB entries need to be updated and entries without valid faces are removed. This mechanism is already available in CCNx since Dynamic Unicast faces are already used in CCNx, i.e., they are created to return Data via unicast when Interests have been received via unicast. However, we reduce the period to check whether the face is still used from 16 s to 4 s to detect path breaks quicker (since path breaks may occur frequently in mobile networks). Consequently, also valid paths are removed quicker if they are not used anymore, limiting the number of active forwarding entries in the FIB. In the worst case, i.e., too early deletion of a FIB entry, another Interest needs to be broadcast to establish a new path to a content source. However, this is not necessarily a disadvantage because it enables requesters to find new (and potentially closer) content sources.

Second, a requester may retrieve different content objects from the same neighbor node via different paths, e.g., from two content sources via a neighbor node as shown in Fig. 5. Then, it is possible that a source, e.g., content source 2, may move away but the neighbor node is still in range to receive and forward messages from and to content source 1. Thus, the requester would still keep the face to the neighbor node (because it receives messages) although content from source 2 cannot be retrieved anymore. This illustrates that automatic deletion of dynamically created unicast faces (case 1 above) is not enough to support route maintenance for multi-hop communication. Consequently, we add a short prefix validity time of 5 s to each dynamically created FIB entry, i.e., slightly more than the default Interest lifetime of 4 s such that retransmissions (in case of collisions) can be satisfied from nearby caches but broken paths still expire quickly. Whenever a Data message is received over the configured face, the lifetime of the configured FIB entry is extended. Otherwise, the prefix is deleted from the FIB after 5 s.

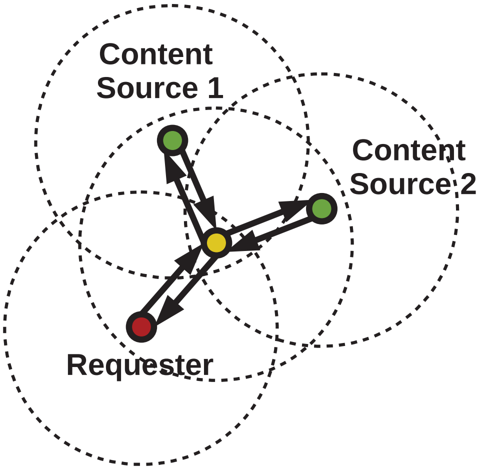


Fig. 5. Retrieving content via neighbor node.

3.5. Interest forwarding strategies

Since multiple faces can be configured for the same prefix in the FIB, forwarding strategies are required to define over which faces Interests are forwarded. We evaluate two forwarding strategies as described below.

3.5.1. Single Face Forwarding (SFF)

This strategy establishes a single path from a requester to a content source. Every Interest is first forwarded over the “best” face and if nothing has been received in return, it is forwarded via broadcast (fallback). If a unicast face is available, it is considered the “best” face (priority over broadcast). If multiple unicast faces are available, the face that has been (successfully) used the last time is considered the best face.

3.5.2. Parallel Face Forwarding (PFF)

This strategy can establish multiple paths between requester and content source. If there are multiple unicast faces, the PFF strategy sends the Interests over all unicast faces in parallel and not only over the “best” face as SFF. Interests are first forwarded via all unicast faces and if nothing has been received (on either face), they are forwarded via broadcast (fallback).

4. Content Request Tracker (CRT)

If there are many concurrent requesters for the same content, e.g., for a video broadcast, a single broadcast transmission may be more efficient than multiple unicast transmissions. Therefore, we introduce a *Content Request Tracker (CRT)* as optional extension of the CCND message forwarding engine. A CRT is a hash table, which maintains one CRT token (struct containing multiple connection parameters) for each actively requested content prefix. A CRT can be applied only at source nodes (CRT-S) or both source and requester nodes (CRT-SR) as described in the following sections.

4.1. CRT at Source (CRT-S)

Fig. 6 illustrates message processing at a content source.

If a content source receives a unicast request, it checks the FIB if the content is locally available in the repository (step 1). If the content is available, the corresponding CRT token is loaded or a new CRT token is created if it does not yet exist (step 2). At the content source, the CRT token maintains information about active unicast connections, i.e., number and node IDs of current requesters, and the time when the last request has been received. The Interest can then be forwarded to the repository (step 3). Before Data can be returned to an individual requester based on PIT information (not shown in Fig. 6), the CRT is consulted (step 4). If a certain number of *MAX_CRT* different unicast requests has been received for the content prefix, the content source transmits the Data message via broadcast (step 5). In this case, all pending Interests in the PIT, i.e.,

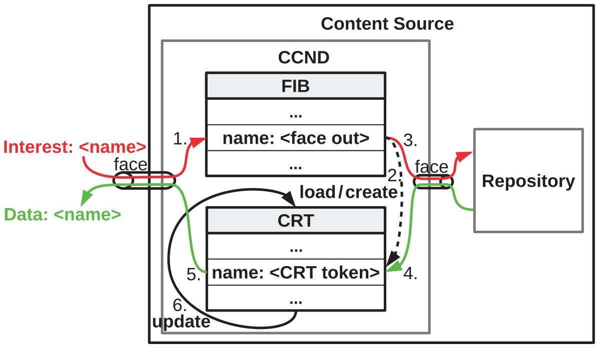


Fig. 6. CRT at source for CRT-S and CRT-SR.

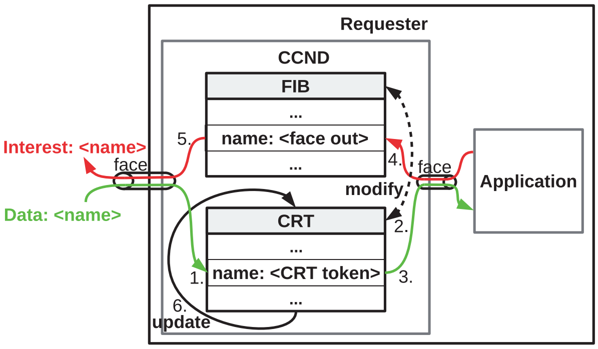


Fig. 7. CRT at requester for CRT-SR.

with unicast faces that would be satisfied by the broadcast transmission, are removed (not shown in Fig. 6).

The CRT is regularly updated (step 6) by deleting CRT entries for content prefixes if no new requests have been received for a time T (in our implementation: 1 min). Furthermore, CRT entries are deleted whenever the content source transmits the final segment of a content object.

4.2. CRT at Source and Requester (CRT-SR)

At the source, CRT-SR uses the same message processing as CRT-S. Fig. 7 illustrates message processing at a requester. After a Data message has been received via broadcast, the requester checks in the PIT whether a unicast request has been transmitted for it (not shown in Fig. 7). Then, the requester checks the CRT for the content prefix (step 1). A requester keeps track of subsequent broadcast responses to unicast requests in a CRT token, i.e., last received segment number and number of subsequent Data messages. Multiple broadcast Data responses would indicate that a content source may have switched to broadcast (CRT-S) due to many concurrent unicast requests. Thus, for a certain number of *MAX_CRT_REQ* subsequent broadcast responses to unicast requests at requesters (in our implementation: MAX_CRT_REQ = 2), dynamically created unicast faces are removed from the FIB (step 2). Then, the Data is forwarded to the application (step 3) triggering the next Interest (step 4), which is transmitted via broadcast (step 5). To avoid fluctuations between unicast and broadcast, new Dynamic Unicast faces are only created at a requester if no CRT token is available for the content name, i.e., the requester has not switched back to broadcast deliberately. The update process of CRT-SR (step 6) is identical to CRT-S.

5. Evaluation

We have implemented Dynamic Unicast (DU) with Single Face Forwarding (SFF) and Parallel Face Forwarding (PFF) strategies as well as CRT-S and CRT-SR in CCNx 0.8.2 [13]. However, all described communication mechanisms can also be supported by CCNx 1.0. In dynamic networks, unicast faces cannot be statically configured due to changing connectivity. Therefore, we compare DU to broadcast communication (as reference) in our mobile scenarios. All evaluations have been performed with NS3-DCE [24] on a Linux cluster [25]. By that, we evaluate the same source code on simulated nodes that would run on real wireless devices. Although this evaluation strategy poses limitations in terms of network size and simulation times, we believe that it increases the credibility and practical relevance of our results.

Table 1
Wireless Configuration.

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
Error Model	Nist error rate model
Energy Detection Threshold	-86.0dBm
CCA Model Threshold	-90.0dBm

5.1. Evaluation parameters

The wireless configuration is 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 unicast data rate is adapted automatically based on the distance between two nodes. Since the broadcast data rate cannot be adapted, it is usually set to the lowest supported rate.

In every evaluation, requesters retrieve a 5MB file (1280 segments à 4096 bytes) from one or multiple content sources. The freshnessSeconds, i.e., how long each segment is valid in the cache, is set to 300 s.

We evaluate wireless (multi-hop) communication in different mobile and static scenarios with multiple requesters, forwarding nodes and content sources. The scenarios and selected topology parameters are explained in the following sections. Since end-to-end paths during multi-hop communication can be disrupted, we use a content retrieval application [26], which persistently stores received segments at a requester. Then, even in case of long disruptions (when cached content may be deleted), content downloads can always be resumed from where they were stopped.

DU establishes a path between requester and content source such that only nodes on the path receive and forward messages (a new path is established if the old path breaks). In contrast, with broadcast all nodes receive messages and decide individually whether they forward them or not. To compare the two schemes we evaluate the message overhead by the following formula.

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

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,

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009

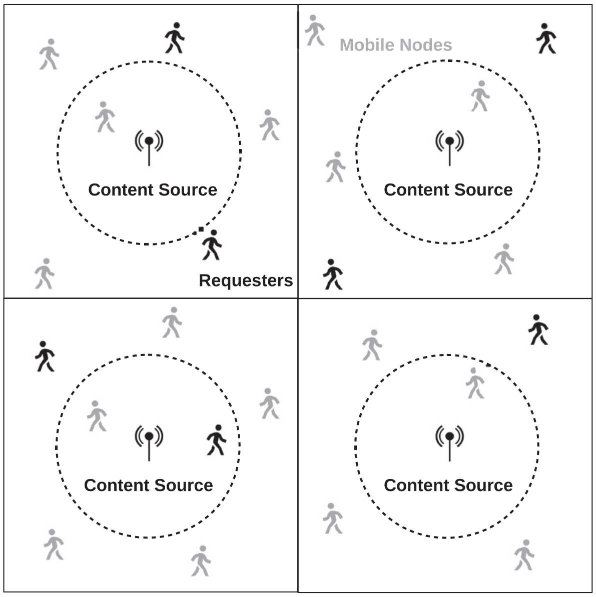


Fig. 8. Grid topology: content sources are placed in a grid, here shown for 4 content sources.

Table 2
Evaluation parameters for grid topology.

Parameter	Value
Nodes	1–36 static sources 1–30 mobile requesters 30 mobile forwarder nodes
Playground	Side length: 1000 m
Mobility	Random waypoint mobility Node speed: 1.2 m/s Node pause: 0 s

requesters and forwarder nodes (neither requesters nor content sources). Every configuration in each scenario has been evaluated in 100 different runs and the boxplots show the average message overhead of all evaluation runs.

5.2. Multiple sources and requesters

Natural disasters, e.g., floodings, earthquakes, or wars may destroy communication infrastructures such that links to central servers are broken. In such scenarios, ICN may enable users to retrieve local emergency information from deployed wireless mesh nodes acting as content sources [27].

Fig. 8 illustrates the evaluation scenario. We use a square playground with a side length $l_{playground}$ of 1000 m and assume that content sources are deployed in a grid. Depending on the number of content sources $n_{sources}$, the playground is divided into smaller regions and each region has a content source in the middle. The side length l_{region} of these smaller regions is calculated by

$$l_{region} = \frac{l_{playground}}{\sqrt{n_{sources}}} \tag{2}$$

For example, Fig. 8 shows 4 content sources that are placed in the middle of regions with side lengths $l_{region} = 500\,\text{m}$.

Table 2 lists the scenario parameters. Within the playground, there are 1–30 mobile requesters, which want to retrieve the same 5MB content concurrently, as well as 30 mobile nodes, which do not perform active requests but can forward received Interests from requesters. We evaluate the performance of concurrent requests in topologies with 1–36 content sources (while 16 content sources are sufficient to ensure one-hop distance to a content source on the entire playground).

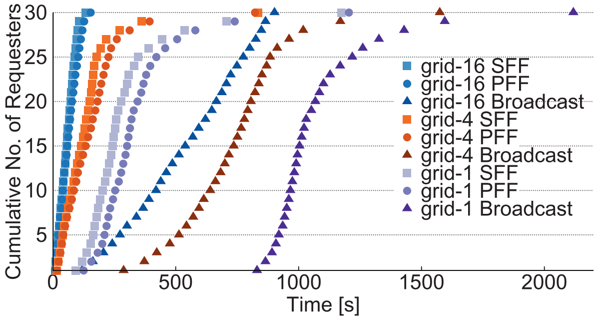


Fig. 9. Cumulative content retrieval times of 30 requesters and different number of content sources.

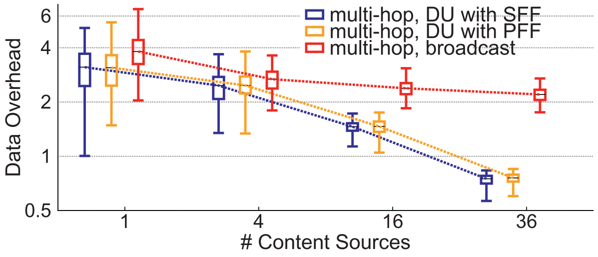


Fig. 10. Data overhead of content sources in case of 30 concurrent requesters.

Fig. 9 shows the cumulative content retrieval times of 30 requesters that retrieve a 5MB file via multi-hop communication. The cumulative content retrieval time denotes the time (*x*-axis) at which a certain number of requesters (*y*-axis) has received the complete file. We only show the results for 1–16 content sources because the results for 36 content sources overlap with 16 content sources. As expected, the content retrieval times decrease with increasing number of content sources. DU is better than broadcast for all source configurations, even when content density is low, e.g., for 1 content source. Although broadcast Data transmissions can be overheard by multiple nodes, which cache the content, content retrieval times via broadcast require more time due to two main reasons. First, DU can exploit short contact times to content sources better due to (potentially) higher data rates and no broadcast delays. Second, overheard and cached content is beneficial in case of multiple requesters (popular content). However, for multiple requesters, the content density is also high with DU. Although requesters cannot overhear and cache unicast Data transmissions from other requesters, they can still cache content, which they requested themselves, resulting in shorter path lengths for other requesters. For all source configurations, multi-path forwarding (DU with PFF) results in slightly worse performance than single-path forwarding (DU with SFF) due to redundant Interest transmissions by mobile nodes.

Fig. 10 shows the transmitted Data messages by content sources for 30 concurrent requesters. Content sources have a significantly lower Data overhead value than 30, which would be the overhead if downloads from 30 requesters would be independent of each other as in host-based communication. Surprisingly, content sources send fewer Data messages with DU than with broadcast despite multiple concurrent requests. While the difference between broadcast and DU is rather low for 1 content source (18% fewer Data messages with DU), the differences increase for higher content densities, e.g., 66% fewer Data messages with DU for 36 content sources. Since DU requests can be addressed to specific content sources (but not multiple content sources at the same time as with broadcast), DU results in fewer Data transmissions

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009

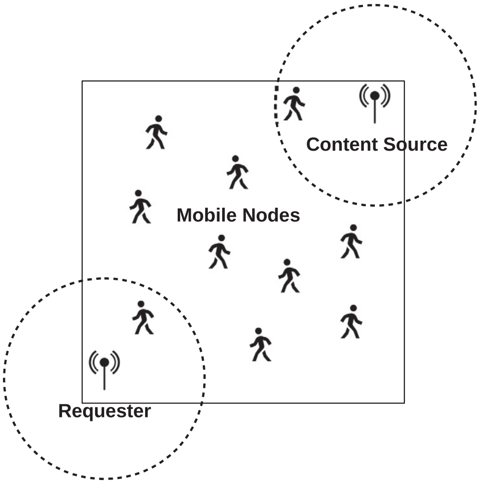


Fig. 11. Static requester and content source with mobile nodes (Forwarders).

Table 3
Evaluation parameters for mobility scenarios.

Parameter	Value
Nodes	1 static source 1 static requester 50 forwarder nodes
Playground	Side length: 374 m Distance: 500 m (source - requester)
Mobility	1. No mobility, static 2. Random waypoint mobility Node speed: 1.2 m/s, 14 m/s Node pause: 0–3600 s

by content sources than broadcast. Despite unicast paths, the Data overhead of content sources for DU is significantly lower than with host-based communication due to in-network caching by requesters and mobile forwarders.

5.3. Mobility during multi-hop communication

In this section, we evaluate the impact of mobility on route persistence during wireless CCN multi-hop communication. Fig. 11 illustrates the investigated scenario. To enforce multi-hop communication, a static requester and a content source are placed at opposite corners of a square playground (10 m to the borders) in 500 m distance to each other. Table 3 lists the evaluation parameters. There are 50 mobile forwarders, which move according to the Random Waypoint mobility model and make occasional breaks, i.e., no mobility. The node pause denotes the maximum break time, e.g., a node pause of 3600 s means that a node randomly waits between 0 s and 3600 s. As reference, we also evaluate a static scenario where static forwarders are randomly distributed in the playground.

Fig. 12 shows the content retrieval times of a requester retrieving a 5MB file from the content source when mobile nodes move with 14 m/s. The *x*-axis denotes the node pause times and the rightmost graphs show the static case (no mobility). For DU with SFF, content retrieval times decrease from a high mobility scenario (node pause: 0 s) to a static scenario by 55% and for DU with PFF they decrease by 60%. However, content retrieval times decrease even with broadcast by 41% from the high mobility (node pause: 0 s) to the static scenario. Hence, we have repeated the same evaluation with a slower node velocity of 1.2 m/s and observed that content retrieval times decrease only by 13% (DU with SFF) and 26% (DU with PFF) from the high mobility to the static scenario.

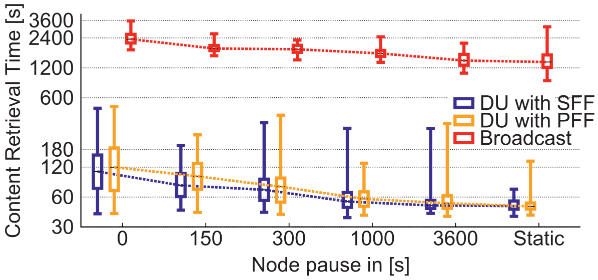


Fig. 12. Content retrieval time for static requester retrieving content from a source at a distance of 500 m. There are 50 mobile nodes (forwarders) moving at a velocity of 14 m/s.

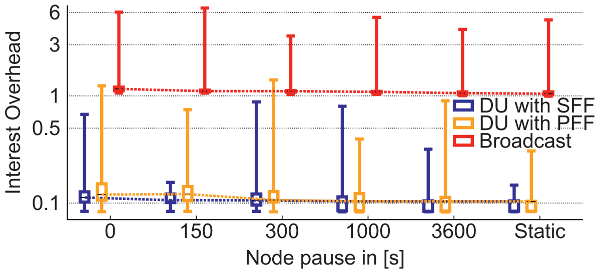


Fig. 13. Interest overhead of mobile nodes (velocity of 14 m/s) during multi-hop communication.

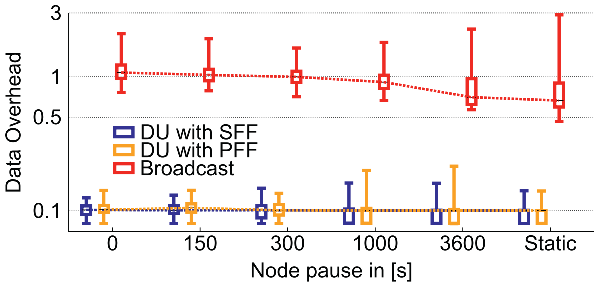


Fig. 14. Data overhead of mobile nodes (velocity of 14 m/s) during multi-hop communication.

Fig. 13 illustrates the Interest overhead by mobile forwarders with a velocity of 14 m/s. DU with SFF results in only 9% more Interest transmissions in high mobility compared to the static scenario, while DU with PFF results in 16% more Interest and broadcast in 11% more Interest transmissions in case of high mobility compared to the static scenario. For a lower node velocity of 1.2 m/s, the increase of Interest messages from a static to a high mobility scenario is significantly lower, i.e., DU with SFF results in only 0.4% more Interests and DU with PFF results in only 1.6% more Interests. This indicates that the main reason for longer content retrieval times in high mobility scenarios are path breaks. In our evaluations, we used the default Interest lifetime of 4 s, which means that Interest retransmissions are only triggered after a timeout of 4 s. However, if Interest lifetimes would be adapted based on estimated round-trip times (RTT), communication may recover faster from path breaks resulting in shorter content retrieval times in high mobility scenarios.

Fig. 14 illustrates the transmitted Data messages by mobile forwarders with a velocity of 14 m/s. We can observe that the number of transmitted Data messages with DU is almost constant, i.e., 0.9% (SFF) and 1.3% (PFF) more Data transmissions in a high mobility compared to a static scenario. This means that breaking of symmetric Interest-Data forwarding paths is not an issue for DU. In contrast, broadcast results in 62% more Data transmissions in high mobility compared to static scenarios. Since broadcast delays

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009

Table 4
Evaluation parameters for multiple requesters.

Parameter	Value
Nodes	1 static source 1–48 static requesters 50 forwarder nodes
Playground	Side length: 374 m Distance: 500 m (source - requesters)
Mobility	1. No mobility, static 2. Random waypoint mobility Node speed: 1.2 m/s Node pause: 0 s

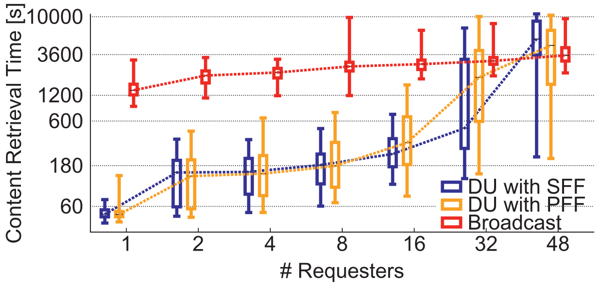


Fig. 15. Content retrieval times of multiple requesters from a source at a distance of 500 m. There are 50 static nodes, which forward the requests.

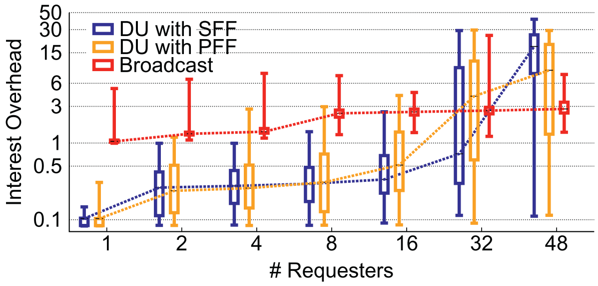


Fig. 16. Interest overhead of static forwarders during multi-hop communication.

(required for duplicate suppression) result in longer (and more varying) round trip times, broadcast forwarding paths are susceptible to mobility.

5.4. Multiple concurrent requests for multi-hop communication

In this section, we evaluate the scalability of wireless CCN communication over multiple hops. We use the same topology as in Fig. 11 but vary the number of static requesters between 1 and 48. The evaluation parameters are listed in Table 4. Besides a static scenario, we also evaluate mobile forwarders with a velocity of 1.2 m/s.

Fig. 15 shows content retrieval times of multiple requesters that retrieve the same 5MB file via static forwarders from the content source. From 1 to 32 requesters, the transmission time increases by a factor of 10.2 for DU with SFF, i.e., less than linear, and only by a factor of 2.2 for broadcast. However, even for 32 concurrent requesters, DU with SFF results in 88% shorter transmission times than broadcast. This observation is consistent with the Interest overhead of forwarders illustrated in Fig. 16. From 1 to 32 requesters, the Interest overhead increases only by a factor of 7 because similar Interests can be aggregated in the PIT. For broadcast, the Interest overhead increases even only by a factor of 2.5 from 1 to 32 requesters, but this is mainly because broadcast Interest transmissions are already on a high level for only a few requesters.

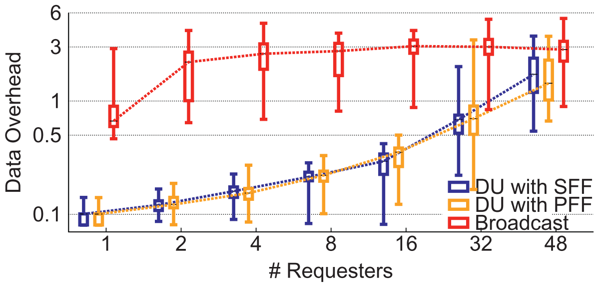


Fig. 17. Data overhead of static forwarders during multi-hop communication.

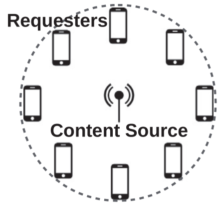


Fig. 18. Multiple requesters placed at equidistance around a content source.

For more than 48 concurrent requesters, DU with SFF results in longer content retrieval times than broadcast (see Fig. 15) because the network is overloaded with Interest messages. In this case, Interests cannot be aggregated efficiently anymore resulting in a significant increase of Interest transmissions (see Fig. 16). In case of mobile nodes, Interest aggregation is less efficient than in the static case (due to changing paths), but even for 32 concurrent requesters, DU with SFF has still 59% shorter content retrieval times than broadcast.

Fig. 17 shows the Data overhead of static forwarders. The Data overhead increases only by a factor of 17 (DU with SFF) from 1 to 48 requesters due to shorter path lengths (caching). We have also observed the Data overhead at content sources and found that for up to 32 concurrent requesters, the content source sends less than 1% more Data messages compared to a single requester and for 48 concurrent requesters, the increase is only 13.5% more Data messages (less efficient Interest aggregation).

In the mobile scenario, the Data overhead of forwarders increases even only by a factor of 14.8 (DU with SFF) from 1 to 48 requesters because some redundant unicast paths (due to multiple requesters) may break more easily with mobility and retransmissions can be satisfied by nearby caches. Since Interest messages are significantly smaller than Data messages, i.e., 50 bytes vs. 4500 bytes, a reduction of Data transmissions has a larger impact on the network traffic than fewer Interest transmissions. Thus, even for 48 concurrent requesters, DU results in lower network traffic than broadcast.

5.5. Content Request Tracker

In this section, we evaluate the performance of a Content Request Tracker (CRT) for multiple concurrent requesters in one-hop distance from a content source as illustrated in Fig. 18. Table 5

Table 5
Evaluation parameters for CRT scenarios.

Parameter	Value
Nodes	1 static source 1–100 static requesters
Playground	Circle around content source Distance: 75 m (source - requesters)
Mobility	No mobility, static

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009

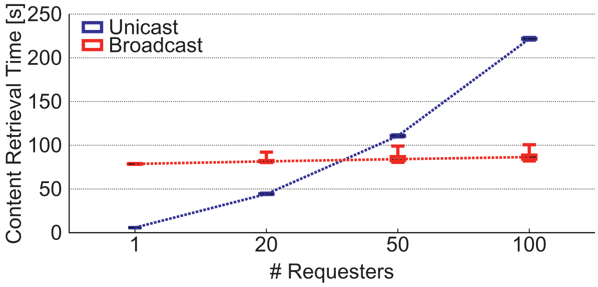


Fig. 19. Content retrieval times for static requesters in one-hop distance.

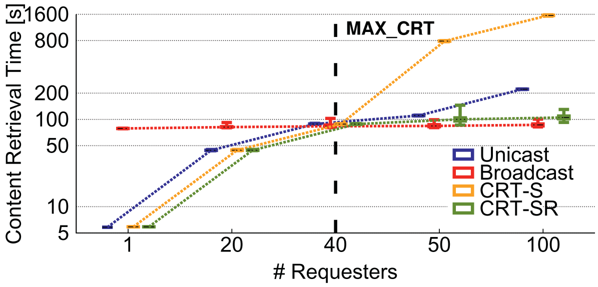


Fig. 20. Content retrieval time for multiple concurrent requesters.

lists the evaluation parameters. Between 1 and 100 requesters are placed at a distance of 75m from the content source.

5.5.1. *Parameter selection*

Different parameters could be selected as decision criterion for unicast or broadcast communication. For example, if the number of transmitted Data messages would be considered, transmissions would always be performed via broadcast for more than 1 concurrent requester. In this work, we select the content retrieval time as decision criterion.

Fig. 19 shows content retrieval times (y-axis) of multiple concurrent requesters (x-axis) for the same 5MB file using broadcast or unicast communication. For only a few requesters unicast content retrievals are faster than broadcast but for more than 50 requesters, broadcast is faster. Based on this observation, we set *MAX_CRT* to 40 concurrent requests. At requesters we set *MAX_CRT_REQ* to 2 subsequent broadcast Data replies to unicast requests.

5.5.2. *CRT-S vs. CRT-SR*

Fig. 20 shows content retrieval times of multiple concurrent requesters for the same 5MB file using unicast, broadcast, CRT-S (CRT only at the content source) and CRT-SR (CRT at content source and requesters).

Surprisingly, CRT-S becomes worse than unicast above *MAX_CRT*. This is because requesters still transmit requests via unicast while the content source responds via broadcast (lower data rate than unicast). Then, if unicast Interests from requesters do not arrive at exactly the same time, the content source may broadcast the same Data messages multiple times as response to each Interest because it does not remember already transmitted Data messages. To improve the performance, requesters need to switch back to broadcast requests as well. CRT-SR performs similar to broadcast for many concurrent requesters, i.e., CRT-SR requires only 22% more time than broadcast for 100 requesters. Compared to unicast, CRT-SR results in 53% shorter content retrieval times for 100 requesters.

Fig. 21 shows the Data overhead of the content source for multiple requesters. Below *MAX_CRT*, the number of Data messages transmitted by a content source using CRT increases linearly with the number of requesters similar to unicast. However, for more than *MAX_CRT* concurrent requesters, content sources with CRT-SR send only 15% more Data messages compared to broadcast because requesters switch to broadcast communication. Thus, the parameter *MAX_CRT* sets an upper bound on Data transmissions with CRT-SR. Lower *MAX_CRT* values may result in lower upper bounds but in longer content retrieval times.

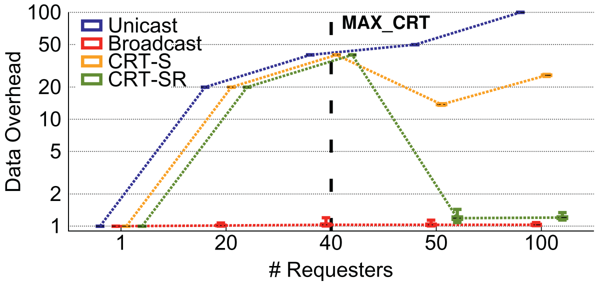


Fig. 21. Data overhead of content source for multiple concurrent requesters.

6. Lessons learned

6.1. *Routing in mobile networks*

Evaluations have shown that Dynamic Unicast with single path routing (SFF strategy) performs better than multi-path routing (PFF strategy) in terms of retrieval times and message overhead. Furthermore, Dynamic Unicast performs better than broadcast even for low content densities because short contact times to content sources can be better exploited. Although broadcast communication is efficient to quickly find a content source, it is not required to perform all communication via broadcast. Particularly for high content densities, Dynamic Unicast shows clear benefits compared to broadcast due to fewer duplicate transmissions.

6.2. *Impact of mobility*

Dynamic Unicast performs best in static scenarios but the performance degrades only slightly if nodes move with pedestrian speeds. However, for vehicular speeds, content retrieval times increase significantly compared to static scenarios. The main reasons for the degradation are path breaks that are only detected after Interests have timed out. While the number of Interest transmissions increases slightly for more mobile scenarios, the number of transmitted Data messages remains nearly constant. Thus, adaptive Interest lifetimes [27] based on measured round-trip times are required to enable more seamless communication in high mobility scenarios. Surprisingly, the performance of broadcast communication decreases also with mobility. In particular, the number of transmitted Data messages during broadcast communication increases significantly in high mobility compared to static scenarios (in contrast to Dynamic Unicast), which indicates that symmetric Interest – Data forwarding paths may break more easily due to broadcast delays.

6.3. *Scalability for multiple requesters*

Evaluations with multiple requesters have shown that Interest messages can be efficiently aggregated in the PIT such that transmitted Interests do not drastically increase with increasing number of requesters (as long as the medium is not overloaded). In case of mobility, Interest aggregation works slightly less efficient since neighboring nodes, i.e., forwarding paths, may change slightly. Furthermore, Dynamic Unicast over multiple hops results in fewer

Please cite this article as: C. Anastasiades et al., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009

Data transmissions than broadcast even for many concurrent requesters in static and mobile scenarios. Due to caching, most requests can be satisfied by intermediate caches and do not need to be forwarded to a content source.

6.4. Replacing multiple unicast transmissions with one broadcast transmission

We have observed that replacing multiple unicast Data transmissions with one broadcast transmission as proposed with the Content Request Tracker is quite complex. It is not enough if only a content source switches to broadcast when requesters still transmit their requests via unicast. However, if requesters switch to broadcast requests as well, a content source has no means of knowing how many nodes are still interested in the content, i.e., whether broadcast is required. Particularly in mobile environments where connectivity changes frequently, the decision whether to use broadcast or unicast cannot be done once but needs to be re-evaluated periodically. Thus, instead of using CRT-SR in combination with Dynamic Unicast, i.e., reverting to broadcast communication in case of multiple requesters, it may be more efficient to avoid Dynamic Unicast, i.e., individual unicast links, for certain content prefixes in the first place, e.g., because the content is of high importance such as in emergency or disaster scenarios.

7. Conclusions and future work

We have explored information-centric routing in mobile and wireless ad-hoc networks. While broadcast is beneficial to quickly find a content source, it is not required to perform all message transmissions via broadcast. Instead, Dynamic Unicast enables requesters to retrieve content from the same content source until it becomes unavailable. We have described two forwarding strategies for Dynamic Unicast as well as an optional Content Request Tracker to enable one broadcast transmission instead of multiple independent unicast transmissions. All mechanisms have been implemented in the CCNx framework and evaluated using NS3-DCE.

Evaluations have shown that CCN can effectively improve scalability of wireless communication because multiple requests can be aggregated and content can be retrieved from caches such that only a fraction of requests needs to be forwarded to content sources. Dynamic Unicast results in significantly shorter content retrieval times and fewer Data transmissions than broadcast for high content densities, but surprisingly it performed also better than broadcast for low content densities.

As future work, adaptive Interest lifetimes based on round-trip times may help to detect path breaks quicker and, thus, improve performance of Dynamic Unicast in case of high mobility. Furthermore, the usage of different MAC protocols can be investigated such as IEEE 802.11p for vehicular networks or new MAC protocol designs tailored for information-centric wireless communication.

References

[1] T. Clausen, P. Jaquet, RFC 3626: Optimized link state routing protocol (OLSR), 2003. https://www.ietf.org/rfc/rfc3626.txt, last accessed: April 2016.
[2] D. Johnson, Y. Hu, D. Maltz, RFC 4728: The dynamic source routing protocol (DSR) for mobile ad Hoc networks for IPv4, 2007, https://tools.ietf.org/html/rfc4728, last accessed: April 2016.
[3] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, LA, USA, 1999, pp. 90–100.
[4] C. Perkins, S. Ratliff, J. Dowdell, L. Steenbrink, V. Mercieca, Ad hoc on-demand distance vector (AODVv2) routing, 2016, https://tools.ietf.org/html/draft-ietf-manet-aodvv2-13, last accessed: April 2016.
[5] B. Ahlgren, C. Dannewitz, C. Imbrena, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Commun. Mag. 50 (7) (2012) 26–36.
[6] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, R.L. Braynard, Network named content, in: Proceedings of the Fifth ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT), Rome, Italy, 2009, pp. 1–12.
[7] J. Wang, R. Wakikawa, L. Zhang, DMND: Collecting data from mobiles using named data, in: Proceedings of the Second IEEE Vehicular Networking Conference (VNC), Jersey City, NJ, USA, 2010, pp. 49–56.
[8] M. Meisel, V. Pappas, L. Zhang, Ad hoc networking via named data, in: Proceedings of the Fifth ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch), Chicago, USA, 2010, pp. 3–8.
[9] M. Amadeo, A. Molinaro, G. Ruggeri, E-CHANET: Routing, forwarding and transport in information-centric multihop wireless networks, Comput. Commun. 36 (2013) 792–803.
[10] L. Wang, A. Afanasyev, R. Kuntz, R. Vuyyuru, R. Wakikawa, L. Zhang, Rapid traffic information dissemination using named data, in: Proceedings of the First ACM Workshop on Emerging Name-Oriented Mobile Networking Design – Architecture, Algorithms, and Applications, Hilton Head Island, SC, USA, 2012, pp. 7–12.
[11] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, L. Zhang, Data naming in vehicle-to-vehicle communications, in: Proceedings of the First IEEE Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN), Orlando, FL, 2012, pp. 328–333.
[12] G. Grassi, D. Pesavento, G. Pau, L. Zhang, S. Fdida, Navigo: Interest forwarding by geolocations in vehicular named data networking, in: Proceedings of the 2015 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Boston, MA, USA, 2015.
[13] The CCNx Project, http://www.ccnx.org/, last accessed: April 2016.
[14] P. Jaquet, P. Mühlethaler, T. Clausen, A. Laouiti, Optimized link state routing protocol for ad hoc networks, in: Proceedings of the IEEE International Multi Topic Conference (INMIC), Lahore, Pakistan, 2001, pp. 62–68.
[15] D. Johnson, D. Malt, Dynamic source routing in ad hoc wireless networks, in: Mobile Computing, Kluwer Academic Publishers, 1996, pp. 153–181.
[16] J. Al-Karaki, A. Kamal, Routing techniques in wireless sensor networks: A survey, IEEE Wirel. Commun. 11 (6) (2004) 6–28.
[17] C. Intanagonwiwat, R. Govindan, D. Estrin, Directed diffusion: A scalable and robust communication paradigm for sensor networks, in: Proceedings of the Sixth ACM International Conference on Mobile Computing and Networking (MobiCom), Boston, MA, USA, 2000, pp. 56–67.
[18] M. Varvello, I. Rimac, U. Lee, L. Greenwald, V. Hilt, On the design of content-centric MANETs, in: Proceedings of the Eighth International Conference on Wireless On-Demand Network Systems and Services (WONS), Bardonecchia, Italy, 2011, pp. 1–8.
[19] M. Amadeo, C. Campolo, A. Molinaro, Enhancing content-centric networking for vehicular environments, Elsevier Comput. Netw. 57 (2013) 3222–3234.
[20] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, L. Zhang, VANET via named data networking, in: Proceedings of the Second IEEE Workshop on Name Oriented Mobility (NOM), Toronto, ON, Canada, 2014, pp. 410–415.
[21] C. Anastasiades, T. Braun, Dynamic transmission modes to support opportunistic information-centric networks, in: Proceedings of the Second International Conference on Networked Systems (NetSys), Cottbus, Germany, 2015, pp. 1–5.
[22] C. Anastasiades, T. Braun, Towards information-centric wireless multi-hop communication, in: Proceedings of the Thirteenth International Conference on Wire and Wireless Internet Communications (WWIC), LNCS 9071, Malaga, Spain, 2015, pp. 367–380.
[23] E. Baccelli, C. Mehlis, O. Hahm, T.C. Schmidt, W. Wählisch, Information centric networking in the IoT: Experiments with NDN in the wild, in: Proceedings of the First ACM Conference on Information-Centric Networking (ICN), Paris, France, 2014, pp. 77–86.
[24] NS-3 Direct code execution, http://www.nsnam.org/overview/projects/direct-code-execution/, last accessed: April 2016.
[25] Ubelix, Linux cluster of University of Bern, http://www.ubelix.unibe.ch, last accessed: April 2016.
[26] C. Anastasiades, T. Schmid, J. Weber, T. Braun, Opportunistic content-centric data transmission during short network contacts, in: Proceedings of the 2014 IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, 2014, pp. 2516–2521.
[27] C. Anastasiades, L. von Rotz, T. Braun, Adaptive interest lifetimes for information-centric wireless multi-hop communication, in: Proceedings of the Eighth IFIP Wireless and Mobile Networking Conference, Munich, Germany, 2015.

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.

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., Dynamic Unicast: Information-centric multi-hop routing for mobile ad-hoc networks, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.03.009