Information-Centric Networking in Mobile and Opportunistic Networks

Carlos Anastasiades[1](✉), Torsten Braun[1], and Vasilios A. Siris[2]

[1] Institute of Computer Science and Applied Mathematics, University of Bern,
Neubrückstrasse 10, 3012 Bern, Switzerland
{anastasiades,braun}@iam.unibe.ch
[2] Department of Informatics, Athens University of Economics and Business,
Patission 76, 10434 Athens, Greece
vsiris@aueb.gr

Abstract. Information Centric Networking (ICN) as an emerging paradigm for the Future Internet has initially been rather focusing on bandwidth savings in wired networks, but there might also be some significant potential to support communication in mobile wireless networks as well as opportunistic network scenarios, where end systems have spontaneous but time-limited contact to exchange data. This chapter addresses the reasoning why ICN has an important role in mobile and opportunistic networks by identifying several challenges in mobile and opportunistic Information-Centric Networks and discussing appropriate solutions for them. In particular, it discusses the issues of receiver and source mobility. Source mobility needs special attention. Solutions based on routing protocol extensions, indirection, and separation of name resolution and data transfer are discussed. Moreover, the chapter presents solutions for problems in opportunistic Information-Centric Networks. Among those are mechanisms for efficient content discovery in neighbour nodes, resume mechanisms to recover from intermittent connectivity disruptions, a novel agent delegation mechanisms to offload content discovery and delivery to mobile agent nodes, and the exploitation of overhearing to populate routing tables of mobile nodes. Some preliminary performance evaluation results of these developed mechanisms are provided.

Keywords: Information-Centric Networking · Mobility · Opportunistic networks

1 Introduction and Motivation

Information Centric Networking (ICN) is a new paradigm for the Future Internet architecture given that the Internet is increasingly used for the dissemination and retrieval of information rather than just interconnecting a pair of particular end-hosts. The most important features of ICN are the usage of content or application-level names/identifiers for addressing, the possibility to cache content

I. Ganchev et al. (Eds.): Wireless Networking for Moving Objects, LNCS 8611, pp. 14–30, 2014.
DOI: 10.1007/978-3-319-10834-6_2

in routers as well as the integrated content discovery mechanisms. Employing content-awareness in the network can help to address a number of limitations in the current Internet's architecture, including mobility support, efficient content distribution and routing, and security.

While most research work investigated the use of ICN in wired networks [1–3] ICN provides some interesting and beneficial features for wireless networks, especially when users are mobile and have rather temporary connectivity with the Internet and between each other, as in opportunistic networking scenarios. This chapter introduces and motivates the usage of ICN in mobile and opportunistic networks and reviews the basic ICN approaches proposed so far. Name resolution (content discovery) and content transfer can be separated as proposed in decoupled ICN approaches or might be integrated as in coupled ICN approaches, each having implications for mobility support.

Related work on ICN and especially for mobile and opportunistic networks is discussed in Sect. 2. ICN concepts nicely support mobility of content consumers, i.e., receivers of content, since no receiver host address information must be updated in case of receiver mobility as it is required in today's Internet mobility solutions such as Mobile IP. However, if content is moving, e.g., when moved from one source to another, or when the source of content is moving, e.g., when content is stored on mobile users' smart phones or on devices located in cars, there are certain issues to be solved. Solutions to address the source mobility problem are extensions of ICN routing protocols, indirection of content discovery messages, and resolution of location-independent identifiers. These are discussed in Sect. 3.

In opportunistic networking scenarios connectivity and contact durations between devices are unpredictable and intermittent. To avoid beaconing and establishing connections among specific end systems, content discovery messages can be transmitted (possibly using broadcast) to find relevant content at neighbour nodes. Section 4 discusses various options for efficient discovery of content on neighbour nodes as well as other issues related to content transfer. To overcome possible connectivity disruptions between devices, we propose to integrate resume functions into ICN, which allows content transfer to continue after connectivity disruptions. Moreover, the chapter discusses the idea of delegating content discovery and retrieval to agents. Also, it investigates the use of unicast and multicast/broadcast for content transfer. Finally, Sect. 5 summarizes and concludes the chapter.

2 Information-Centric Networking

2.1 Coupled and Decoupled ICN Approaches

Information-Centric Network (ICN) architectures depart from the current Internet's host-centric end-to-end communication paradigm and adopt an information (or content) centric communication paradigm, where information objects, rather than host end-points, are named. Receivers (or subscribers) request information objects by their names and the network is responsible for locating the sources (or publishers) of the information objects and transporting the objects from the sources to the receivers. Three key functions of Information-Centric Networks are the following:

- ***Name resolution*** involves resolving (or matching) the name of an information object with its location or its source. Name resolution can be performed in a hop-by-hop manner or by an independent name resolution system. The name resolution system can have a hierarchical structure: subscribers and publishers communicate with a local name resolution server, which in turn communicates with other name resolution servers if necessary.
- ***Topology management/routing*** involves determining a path from the source to the receiver. Different domains can implement different topology management and routing procedures. Similar to the name resolution system, topology management can be performed in a hierarchical manner.
- ***Forwarding*** involves moving information from the sources to the receivers along the determined path. Possible forwarding mechanisms include hop-by-hop forwarding based on end-system IDs, label switching, and forwarding based on a series of link identifiers selected by the source.

Different ICN proposals involve a different degree of coupling between name resolution and routing/forwarding [4]. At one extreme (tight coupling), the same network nodes perform both functions in an integrated manner. This is the approach followed by Content Centric Networking (CCN)/Named Data Networking (NDN) [5,6]: Receivers express their request for content using Interest packets, which serve for content discovery. Such Interest packets are routed based on the name of the requested content, using longest prefix matching, either to the source that contains a data packet with the requested name or to an intermediate network node that has cached the requested data packet. Once the data packet is found, it is returned to the requester following the reverse path of the received Interest packet.

At the other extreme (decoupled), the functions are implemented in different network nodes and/or different modules. This is the approach followed by architectures such as PSIRP/PURSUIT's PSI (Publish-Subscribe Internet) [4,7] and 4ward/SAIL's NetInf architecture [1,8]. With such an approach, the name resolution system is independent and operates as an overlay of the routing/forwarding network, which transfers content from the source to the receiver. This has similarities with the current Internet's Domain Name System. Proposals such as DONA [3] and COMET [9] describe overlay solutions that run on top of an IP infrastructure, hence inherit IP's routing and forwarding functionality. A more detailed survey and comparison of the similarities and differences of the most important ICN proposals can be found in [4].

Decoupling the resolution and routing/forwarding functions allows more flexibility in where and which entities implement this functionality. This flexibility can allow existing or new mechanisms, e.g., for routing and forwarding, to be used in different domains that have specific characteristics or restrictions, such as satellite networks or home networks. Decoupling allows usage of separate paths for control traffic and data traffic. Moreover, data transfer can utilize multiple paths (multi-path) from one or more information publishers (multi-source) to a subscriber. Another property of decoupling name resolution and routing/forwarding is that the resolution layer can employ a receiver-driven (pull-based) communication mode, whereas the routing/forwarding layer can employ either a receiver-driven (pull-based) or sender-driven (push-based) communication mode. This, for example, allows a receiver to declare (through a subscription message) its interest in receiving future content related to some content category. Once the publishers (sources) create such content, they can send it (push-based) to the receiver without requiring requests for each individual content object. On the other hand, when resolution and routing/forwarding are coupled, then implementation of sender-driven (push-based) functionality requires either overlay solutions to inform receivers of the availability of content, or polling-based solutions where the receivers periodically poll the sources for new content. It is interesting to note that for architectures that employ a similar level of coupling between name resolution, topology control/routing, and forwarding, the same mechanisms and algorithms can be implemented for the same functionality.

2.2 Related Work in ICN for Mobile and Opportunistic Networks

CCN in mobile networks has already been the subject of several studies [10]. Early works investigated the applicability of existing MANET routing protocols for mobile CCN based on analytical models [11]. A hierarchical CCN routing scheme based on distributed meta information has also been implemented [12]. The Listen First, Broadcast Later (LFBL) [13] algorithm limits forwarding of interests at every node based on its relative distance to the content source. However, all these works assume continuous network connectivity and do not consider intermittent connectivity.

Opportunistic and delay-tolerant communication has been investigated extensively in the last decade. The Bundle Protocol [14] describes a delay-tolerant protocol stack to support intermittent connectivity. The destinations of messages, i.e., bundles, are identified by endpoint identifiers. To receive bundles, nodes can register in endpoint identifiers and these registrations are exchanged when two devices meet. Thus, bundles are transmitted in bursts and stored locally until the next forwarding opportunity arises. Haggle [15] describes a data-centric network architecture for opportunistic networks. The platform uses device discovery to establish point-to-point connections between devices. Data is described by meta data composed of multiple key words. Users express and forward interests containing keywords when connected to other devices. All data objects that match the keywords are forwarded to the requesting node by a push-based dissemination model. The successor project of Haggle, called SCAMPI [16], developed a service-oriented platform for mobile and pervasive networks, which benefits from opportunistic communication paradigms. Routing and opportunistic networking is hidden from applications through a middleware. It contains a communication subsystem, which is responsible for detecting neighbouring peers and exchanging messages. Direct peer sensing mechanisms are applied to discover peers and services within communication range based on IP multicast or static IP discovery. To discover nodes further away, the platform defines transitive peer discovery, where nodes exchange information about other nodes they have discovered. Routing of messages in the network is based on discovered peers and controlled by the routing subsystem.

CCN can support opportunistic networking without device (or peer) discovery because data transmissions are based on content names available in the current environment. Investigations [17] already identified the potential of CCN for delay-tolerant networking (DTN).The effectiveness of CCN for opportunistic one-hop content discovery has been investigated in an earlier work [18]. There are also related efforts in creating a new content-centric opportunistic networking architecture inspired by CCN [19].

3 Mobility Support in Information-Centric Networking

This section considers mobility support in ICN architectures in more detail. Mobility support is particularly important within the context of moving objects and things that are network connected. Receiver mobility and sender mobility are discussed separately, since they have different requirements and can involve different mechanisms.

3.1 Receiver Mobility

ICN architectures promote a receiver-driven information request model, where nodes receive only the information which they have requested or subscribed to. This is in contrast to the current Internet's model, where the sender has full control of the data he/she can send. Additionally, ICN's request model and content transfer from sources to receivers is connectionless, in contrast to TCP's connection-oriented (stateful) end-to-end control, which involves location-dependent addresses. Both the above features allow mobile devices that have changed their positions (network attachment points) to simply re-issue requests for information objects they did not receive while they were connected to their previous attachment point or while they were disconnected [20]. Hence, delay/disruption tolerant operation in addition to mobility is supported without requiring cumbersome solutions such as Mobile IP.

Specific schemes for enhancing mobility support have also been proposed in the context of ICN architecture proposals. In rendezvous-based schemes the rendezvous service has the major role. The (moving) receiver upon re-location and re-attachment to the network needs to re-issue a subscription for the content he/she did not receive due to their movement. Upon receipt of this subscription the rendezvous service returns the new path for connecting the receiver with a sender (either the same or a new one). Depending on the service (streaming or file transfer), lost packets (those that were being transferred during the hand-off) may need to be recovered or not. If packets need to be recovered, then the new subscription may also contain a hint about the last successfully received chunk of content, which the rendezvous service communicates to the newly chosen source, so that lost packets can be recovered. If it is useless to recover packets (e.g., if the subscription is for a real-time video stream and the play-out time for the frames contained in the lost packets has passed) then the rendezvous service simply returns the new path from the source to the re-located receiver. Some approaches, such as CCN/NDN, require that subscriptions (or interests) have to be issued for every packet, so in this case the receiver upon re-attaching to a new location simply re-issues the non-satisfied interests. In addition to the inherent support for mobility, additional mechanisms, such as proactive caching [21], can be further utilized to reduce the delay for obtaining time-critical information.

3.2 Source Mobility

Unlike receiver mobility, source mobility in ICN architectures requires additional mechanisms. In particular, the following two issues need to be addressed with source mobility: (a) find the source's location, which includes finding the source's location in the beginning of communication but also tracking the source when it moves, and (b) session continuity, which involves reducing the impact of mobility, such as reducing disconnection periods, minimizing/avoiding data loss during mobility, and supporting graceful disconnection and fast reconnection.

How source mobility can be supported depends on whether name resolution and data transfer are coupled or decoupled. In CCN/NDN, where name resolution and data transfer are coupled, receivers issue Interest messages, which contain the name of the requested content; these Interest messages are routed towards the source based on FIB (Forward Information Base) entries. If the source changes its location, it will need to issue a new prefix announcement from its new location. These prefix announcements are distributed (e.g., flooded using a link state protocol) to other CCN/NDN nodes in the network, which update their FIB tables. Note that the above approach can have some similarities with service advertisement and discovery; specifically in wireless networks, the broadcasting nature of the wireless channel can be used to advertise services locally. Nevertheless, updating routing information in large networks with multiple domains requires mechanisms for disseminating location information across multiple domains.

On the other hand, in architectures where name resolution and data transfer are decoupled, source mobility requires updating the resolution information, which maps names to locators. In cases where multiple sources offer the same content as the source that moved, the rendezvous service may also choose to assign some (or all) of the receivers that were served by the source that moved to other new sources that are closer to the receivers.

There are three approaches for supporting source mobility: (1) the routing-based approach, (2) the indirection approach, and (3) the resolution approach.

The ***routing-based approach*** involves updating the routing tables that are used to forward information requests, as in the case of CCN/NDN. Issues with this approach include convergence time and scalability of the routing tables. The approach can be enhanced to reduce the routing convergence time by using a proactive prefix advertisement scheme, similar to the proposal in [22]. Additionally, if mobility prediction information exists, then proactive actions along the lines of [21] can be utilized.

The ***indirection approach*** is based on home agents, which forward interests to the mobile device and are updated with the mobile device's current point of attachment, similar to Mobile IP. The approach also requires agents in the visited network and location-based identifiers. An advantage of this approach is that there is no overhead due to a resolution phase. A disadvantage is that all requests and data packets go through the home agent. Moreover, home agents would require names with topological/location information to be able to forward requests to the mobile source.

The ***resolution approach*** involves a separate resolution phase: the receiver first sends a request containing the name of the requested content; it gets a response containing the location-dependent name or address to use for obtaining the requested content. Hence, this involves a location-identity split and has similarities with HIP (Host Identity Protocol) [23]. The resolution approach adds overhead, which however is limited to the first packet. Also, this approach requires some form of agent in the visited network. If a name resolution function already exists, then the approach can be implemented by updating the name to locator binding that is used for resolution. On the other hand, if a resolution system does not exist, such as in CCN/NDN, then a resolution phase can be added in two ways: use names that contain location/topology information [22] or add a locator field in Interest messages [24].

The source mobility solution depends on whether the particular ICN architecture supports only names (location independent identifiers) or both names and locators (location or topology dependent addresses). If only names are supported, then only the routing-based approach can be applied. If names are generic, then some form of location dependence can be added to names, e.g. exploiting some hierarchical structure of names [22]. In this case, the indirection approach can be applied. However, note that the addition of location dependence to names can have implications to mechanisms such as in-network caching and content-aware processing, which assume that names are location independent. If both names and locators exist, as advocated by recent proposals [2], then the resolution approach offers higher flexibility.

4 ICN in Opportunistic Networks

Opportunistic networking defines communication in challenged networks, where connectivity and contact durations between devices are unpredictable and intermittent. The main goal is to exploit contact opportunities between users to support best-effort content and service interactions when fixed network infrastructures may not be available. Based on exchanged beacons, users detect neighbouring devices as communication opportunities and need to connect to neighbours individually to perform content discovery and file transmissions. ICN can support opportunistic networking because all communication can be performed inside the local network environment. No device discovery is required, because content availability may be independent of neighbouring devices. Mechanisms for content discovery are described in Subsect. 4.1, whereas content transfer techniques are discussed in Subsect. 4.2. All investigations are done using a CCN/NDN implementation based on the CCNx software [25].

4.1 Content Discovery

Nodes need to transmit requests, i.e., Interest messages, to find available Data. Therefore, transmissions are only performed if there is a node interested in it. If a requester moves within the network, the recipients of Interest messages will change and new content sources or forwarders are discovered automatically. Content discovery is performed using multicast to quickly detect nearby available content sources. If a multicast Interest message is not answered by a neighbouring node, no matching content is available, which - in terms of content retrieval - is equivalent to the unavailability of neighbouring devices. Content discovery is required in distributed environments without centralized directories to learn about available content or service options without demanding the content or service completely. In the following we describe two different discovery approaches and an extension to ensure flexibility of content discovery.

Discovery Algorithms. In opportunistic ICN, we assume that content names follow a hierarchical structure comprising multiple name components. Each data file consists of one or several segments similar to chunks in Bittorrent. The hierarchical name structure may not indicate the location of content objects and content may be stored on one or multiple hosts. The first name component may be based on the identity of a content publisher and the following components are arbitrarily chosen based on the publisher's naming scheme. We designed two discovery mechanisms: *Enumeration Request Discovery* and *Regular Interest Discovery* [18].

Enumeration Request Discovery (ERD) requires the expression of enumeration requests, which are addressed to local and remote repositories only. A name enumeration request for a certain prefix */A* requests next level components that are available in a repository, e.g., $\{a1,\ b1,\ c1,\ d1\}$. To discover the entire name space, the algorithm starts from the top of the name tree with the shortest possible prefix and sequentially moves down to the leaves by extending the prefix with the discovered name components, e.g., */A/a1* in the next step. At every iteration and level, the requesting user receives a list of available next-level components at a specific repository. We assume that mobile repositories are not synchronized among each other. Therefore, the requesting user has to address each repository that holds content with a specific prefix separately until no information is received anymore, i.e., timeout event.

Regular Interest Discovery (RID) is based on recursive expression of regular Interest messages. The user expresses an Interest in a prefix */A* and receives the first data segment of a content object in response, e.g., */A/B/C/segment_0*. Although this leads to overhead because only the content name and no data is required, it is still more efficient than retrieving all segments in complete file downloads. The requester knows the complete name of a content file at the leaf of the tree after only one content request and can sequentially browse its way up to the root. In contrast to ERD, where every component list is unique due to the repository that created it, although it may contain the same information, duplicate content transmissions can be identified and suppressed.

Evaluations have shown that multicast discovery is advantageous in wireless environments, because it addresses multiple content sources simultaneously. If nearby content sources provide different diverse content, a single discovery Interest can pull multiple content objects at the same time. Only exact duplicates of the same content are suppressed. ICN systems enforce one-to-one relations between Interest and Data messages. However, in wireless networks it is beneficial to keep unsolicited content for a short time in the cache so that it can be retrieved in follow-up requests resulting in fewer transmitted messages and higher discovery efficiency.

Two delay values are important for multicast discovery: *transmission delay (TD)* and *requesting delay (RD)*. The transmission delay defines the transmission interval $[\mathrm{TD}, 3{\cdot}\mathrm{TD}]$ within which each host randomly selects a time to reply with a content object. Once scheduled, the content object stays in the senders' send queue until the transmission delay is due enabling duplicate suppression by removing overheard content from the send queue. Larger TD values result in fewer collisions and duplicate content transmissions but increase discovery time. However, even large TD values result in non-negligible duplicate content transmissions. To reduce duplicate content transmissions to nearly zero, a requesting delay of $2{\cdot}TD$ is required. The requesting delay defines the delay between subsequent Interest requests and equals the maximum difference in the transmission interval $[\mathrm{TD}, 3{\cdot}\mathrm{TD}]$ to ensure that another response is received and can be found in the cache without additional re-expression. If a requester transmits the next request quicker, not all answers from other content sources may have been received yet. If the next Interest arrives at a content source just after it has answered the previous request, the same content object will be returned since content sources do not memorize recently transmitted content.

ERD is independent of the number of content objects but it depends on the number of content sources and may, therefore, be inefficient in mobile networks where neighbours change frequently. Compared to RID, the ERD content lists of all repositories need to be processed and accumulated to know which content names are available. If all hosts store the same content, ERD requires all nodes to request and process all content lists without learning anything new. RID is more efficient to detect small differences in collections, because it can ask specifically for new content: redundant information can be avoided via duplicate suppression. RID is also faster in finding content names in highly structured name spaces with many name components where ERD would require subsequent traversing through all name components until reaching the content objects. Therefore, a combination of both approaches may be promising: an initial RID request may quickly find the full name of a content object and subsequent ERD requests discover all available name components on the same level.

Alias Mapping. To support communication during short opportunistic contacts, it is important that nodes discover available resources quickly. The hierarchical name structure may not be flexible enough to support location-based discovery. Hereafter, we describe how CCN can be used to detect local services or content independently of the publisher that provides it but based on local context. This can be achieved with temporary broadcast names that can be mapped locally to available unique names, i.e., alias mappings.

Broadcast components can be temporarily used by many publishers to describe content, e.g., via */<content_description>/<node_type>/ <node_Id>/*, and are not bound to the public key of a specific unique publisher so that everybody can publish within the name space *content_description/node_type*. Published content objects are signed by the corresponding publishers. A sample data name that follows that structure is */temperature/sensors/sensor_A/*. If connectivity to sensor_A breaks, requesters can quickly find alternative sensors in the vicinity by shortening the prefix to */temperature/sensors/* addressing all nearby sensors that provide temperature.

Alias mappings map names that use broadcast components to unique names. To ensure flexibility in the content description, a content source may map multiple broadcast components to the same unique content name or a list of locally available unique names. For example, a sensor node may use the broadcast components */weather* and */temperature* for the same content. To identify redundant content transmissions for multicast and ensure efficient storage, name aliases link broadcast components to unique names in the form */node_Id/name*. Subsequently, the unique content name is used during data transmission enabling duplicate suppression because the same content can be identified.

4.2 Content Transfer

Content transmissions are only performed in response to a received Interest message. A requester needs to transmit Interests in every segment to receive the complete content. By that, content transmissions are only performed in the vicinity of an active requester. Received segments are first included in temporary cache (content store) and complete files are stored on persistent storage (repository). This subsection describes a cache extension for intermittently connected networks and introduce an agent-based approach that can be used if requester and content source would never meet or to increase content density. Finally, the benefits and disadvantages of multicast communication and overhearing are discussed.

Resume Functions for Intermittent Connectivity. During short opportunistic contacts and, hence, intermittent connectivity between content requesters and content sources, file transmissions may not be completed. If no alternative content sources are available, content is kept in the requester's cache until it can be completed and properly stored. Unfortunately, persistence of data in CCN caches is not guaranteed since they are limited in size and can be overwritten by other files depending on the cache replacement strategy. Caches are built upon high-speed memory to support quick forwarding. In delay-tolerant networking, memory speed is not important since delays between successive requests are high. Therefore, in case of disruptions, partial data can be stored on and loaded from secondary storage.

Content-centric overlays to existing DTN protocols such as the Bundle protocol would experience multiple drawbacks. First, multiple Interests would be required to obtain all segments. Since requesters do not know the file length until receiving the last segment, proactive transmission of Interests would be required. If Pending Interest Table (PIT) entries are valid for a long time as required in DTN networks, the PIT size would drastically degrade lookup performance. Second, long-living Interests prevent forwarding of similar Interests for the entire lifetime period even if the environment has changed and the content becomes available. Therefore, the Interest lifetime should be limited to a rather small value but Interests can be re-expressed periodically to account for changes in content availability.

Every segment is named individually with a segment number. Thus, disrupted downloads can be precisely resumed from where they were stopped. For every incomplete and aborted file, the received partial data is stored in the file *name.part* and the meta information in the file *name.meta*. The meta information includes name and version of the content, the segment number that is expected next, the file position in the partial file *name.part* and the publisher's public key digest. To avoid incomplete files that never get completed or storing data of real-time traffic, an expiration time indicates a timeout value after which the partial files can be deleted. The expiration time can be based on the *reception time* and the *freshnessSeconds* values of the first received segment. In case of real-time traffic, content would only be valid for a few seconds and no partial information would be stored. While strategies without resume operations may never be successful, resumed file transmissions result in constant effective transfer times independent of the time they where disrupted.

Evaluations on wireless mesh nodes showed that the processing and storage overhead is negligible and does not affect file transfers without disruptions in any way. If content sources are unknown, transfers need to be performed by multicast because no unicast addresses can be statically configured. Unfortunately, multicast transfer rates are considerably lower than unicast rates. Additionally, no MAC layer acknowledgements are transmitted during multicast communication. Thus, missing segments, e.g., due to collisions, are detected only after the Interest lifetime has passed and the Interest is re-expressed. However, since opportunistic communication is performed via one hop, the Interest lifetime can be decreased to a lower value to reduce retransmission delays. Evaluations have shown that this strategy can increase the multicast throughput by a factor of 7.2 without significantly increasing the number of transmitted messages [26].

Transfer Agents. In situations where a requester never meets a valid content source, it cannot request content. The proposed solution for this problem is agent-based content retrieval, where requesters can delegate content retrieval to agents, which retrieve content on behalf of the requesters. Communication exploits the agents' mobility. The approach comprises three phases. In the agent-delegation phase (phase I), the requester needs to find an agent and delegate content retrieval to it. In the content retrieval phase (phase II), the agent is looking for the content and retrieves it. In the notification phase (phase III) the requester asks available agents whether they retrieved the complete content. The requester can then retrieve the content from the agent node.

Phase I: Agent Delegation. If a requester cannot find the desired content in its environment, content retrieval can be delegated to an agent. In phase I, the requester finds and assigns an available agent based on a three-way handshake protocol. An agreement between requester and agent can be enforced by signing the exchanged Interest and Data messages with the sender's private key so that both nodes know the identity of each other. Because available agents in the neighbourhood are not known and can change, agent discovery and delegation is performed via multicast. The requester transmits first an *Exploration Interest* in the name space */ferrying/%C1.<namespace>∼<param>*. Every agent application listens to Interests for */ferrying* followed by the *namespace* of the content to be found and optional additional parameters. Parameters may describe an area where content retrieval should be performed and agents can decide whether to respond based on locally collected mobility traces. Agents return an *Exploration Response*, which is a Data message including the requested prefix name and appending their */nodeId* at the end. Exploration Responses have short lifetimes of only a few seconds to avoid usage of old information from the cache. Since Exploration Interests are transmitted via multicast, they trigger potentially many answers and the requester can subsequently poll its content store for other responses. The requester can then create an agent list that includes all available agents and selects one from it for delegation. The requester assigns an agent by transmitting a *Delegation Interest* with the name prefix */ferrying/nodeId/%C1.<namespace>∼<param>/rTime/groupId.* The nodeId is included right after the /ferrying prefix so that all nodes receive it and know whether they have been selected or not. *rTime* defines the remaining time, i.e., how long the requester is still interested in the content. This is an upper limit for content retrieval and after this time has passed, the agent does not look for the content anymore. *groupId* is a random nonce, which is created by the requester for every delegation in order to create a multicast group of agents. Assigned agents will listen to Interests with the */groupId* to receive notification requests from the requester in phase III.

Phase II: Content Retrieval. After receiving the Delegation Interest in Phase I, the agent registers the */namespace* to the multicast face using a lifetime based on *rTime*. Then, it can probe the environment for the availability of a content source similar as explained above for resumed transfers. An agent needs to replicate the received content including all CCN header information and original signatures so that the requester can verify that the content is authentic and produced by the original publisher. Therefore, as soon as connectivity to a content source has been detected, the agent delegates content retrieval to its mobile repository, which is an application running on the same device as the agent. The repository can then request all content objects via the multicast face. When the content transfer is complete, the agent can answer notification requests in phase III.

Phase III: Notification and Content Distribution. Since notifications can only be transmitted in response to Interests, the requester needs to request content notifications from any agent in the vicinity that has retrieved a content object completely. The pull-based approach is advantageous in mobile networks with multiple agents. Since only requesters periodically ask for notifications instead of multiple agents transmitting beacons, fewer notification messages need to be transmitted, i.e., only in the requester's vicinity. The Notification Request is an Interest message with the name */groupId/namespace* and is transmitted periodically until a *Notification Response* is received. By using the groupId, all assigned agents in the requester's transmission range receive the request and only agents that have completed phase II will respond by a Notification Response, which is a Data message that uses the same name as the Notification Request. The payload of the Notification Response comprises the current IP address of the mobile agent so that the requester can create a unicast face to the agent's mobile repository. The IP address can be viewed as locator of the content, which is not part of the routable prefix included in Interest packets. After the requester creates a new unicast entry with a short lifetime, the content can be requested directly via unicast from the mobile repository.

Evaluations on Android smart phones showed that the overhead for agent-based content retrieval compared to two hop forwarding can only be measured for very small files of 1 MB or less. For files larger than 4 MB, agent-based content retrieval resulted in 20 % higher throughput than with two-hop forwarding although content is stored at intermediate nodes on secondary storage but not in the cache. Because the maximum number of concurrently transmitted Interests is limited by the pipeline size, the overall transfer rate during multi-hop forwarding is limited by the slowest link. This means that transmissions via unicast on the first hop can never exceed multicast throughput on the second hop. With agent-based retrieval, content is transmitted subsequently via multicast and unicast over both hops, and, thus, every link can reach its maximum capacity. Moreover, multi-hop forwarding over multiple hops may not be possible (or only at very low rates) due to intermittent connectivity between the network nodes [27].

Multicast and Overhearing. CCN Forwarding is based on registered prefixes in the Forwarding Information Base (FIB). In opportunistic networks, forwarding tables cannot be configured statically and multicast communication is required to find suitable content sources. Since topologies can change, proactive exchange of all content information from all nodes may overload the network. In static networks or networks with limited mobility, delegating content retrieval to one-hop neighbours as described above may not be enough. In such situations, it is required to forward Interests over multiple hops to discover content on nodes, which are just outside the requesters' transmission range.

To enable wireless multi-hop communication in content-centric networks, it is required to configure two faces, one for receiving and one for transmitting. Interests can then be forwarded via alternating faces, e.g., via face 2 if received on face 1 and vice versa. Nodes that overhear content transmissions on a specific face can include this information in the FIB to forward Interests received from others to this face. To ensure that every node can always try to request content from its neighbours and, thus, enable overhearing of Data, Interests from local applications need to be forwarded via a multicast face if no matching FIB entry is configured, i.e., pass-through. If the content is available at a neighbour, the requester overhears the response and registers this information in the FIB. After that, the requester can also forward Interests received from other nodes to the content source via the newly created FIB entry. If the content is not available at neighbours, and neighbours have no FIB entry configured, they discard the received Interest to avoid unbounded forwarding. The registered FIB entry is valid for a limited time but is updated with successive content receptions.

To limit update and processing operations, only every *nth* received content object is processed. By using modulo operations on the received segment number included in the name, only the *1st* and every *(n+1)-th* content object during a transmission is processed not requiring the maintenance of additional state information. Processing the *1st* object is important since it indicates that a content source is available. Every other *nth* content object results in a FIB update. Entries that are not updated will expire automatically after time.

Evaluations on wireless mesh nodes have shown that processing overhead is only measurable for very small files and becomes negligible for files of 10 MB or larger. This is because in the beginning of every file transfer when receiving the first segment, a new FIB entry needs to be included in the FIB. Later, only lifetime values of existing FIB entries need to be updated. During large file transfers, the ratio of entry updates vs. entry creations increases and, therefore, the relative processing overhead decreases. Since overhearing is a passive activity, it has no impact on the number of transmitted messages. Energy measurements revealed that a content source has the largest power consumption but the power overhead is only marginal. A requester with enabled overhearing functionality has a 1.3 % higher power consumption, because received prefixes need to be extracted and included in the FIB. The largest power overhead of 4.6 % is measured at a passive listener, which moves from a passive to a more active role when adding prefixes to the FIB. Compared to the unicast case, where passive nodes can stay in idle mode, the energy overhead of passive listeners not participating in the communication is 22 % higher.

However, overhearing can help to increase multi-hop throughput since retransmissions can be performed quicker from cache, similar to [28]. Additionally, in contrast to caches, which are limited in size and where cache replacement strategies make content disappear quickly, the RAM memory requirements for a content name is considerably lower than for the full content. For example, a 5 MB file may require 5.5 MB storage in the cache, i.e., including CCN headers and signatures, whereas the FIB entry requires only a few bytes depending on the length of the content name. Therefore, FIB entries based on overheard content names indicate a node whether it is worth to forward received Interests to a nearby content source if the cache entry has already been cleared. A combination of multicast discovery and subsequent unicast data transmissions after the content source has been identified may further improve energy efficiency and throughput but it requires modifications of the current CCNx daemon.

5 Conclusions and Outlook

This chapter discussed the use of ICN in mobile and opportunistic networks. Although ICN in such environments seems to have huge potential and benefits, there are still some problems and issues to be optimized. In mobile networks, the problem of source mobility is challenging and requires an appropriate solution for scenarios where content is located in mobile user devices or at devices deployed in vehicles. Moreover, the chapter discussed several problems in opportunistic network scenarios such as the problem of intermittent connectivity between devices or the problem that two devices might never be in contact to each other. Solutions such as suspending and resuming content transfer between devices and delegating content retrieval to agents have been proposed and preliminary evaluations in small-scale testbed scenarios have been performed.

References

1. Ahlgren, B., Aranda, P.A., Chemouil, P., Oueslati, S., Correia, L.M., Karl, H., Sollner, M., Welin, A.: Content, connectivity, and cloud: ingredients for the network of the future. IEEE Commun. Mag. **49**(7), 62–70 (2011)
2. Baid, A., Vu, T., Raychaudhuri, D.: Comparing alternative approaches for networking of named objects in the future internet. In: Proceedings of IEEE INFOCOM NOMEN Workshop, March 2012
3. Chun, B., Ermolinskiy, A., Kim, K.H., Shenker, S., Stoica, I.: A data-oriented (and beyond) network architecture. ACM SIGCOMM Comput. Commun. Rev. **37**(4), 181–192 (2007)
4. Xylomenos, G., Ververidis, C.N., Siris, V.A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K.V., Polyzos, G.C.: A survey of information-centric networking research. IEEE Commun. Surv. Tutor. **16**(2), 1024–1049 (2014)
5. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M., Briggs, N., Braynard, R.: Networking named content. In: Proceedings of ACM CoNEXT, December 2009

6. Named Data Networking project. http://www.named-data.net/index.html
7. PURSUIT Project. http://www.fp7-pursuit.eu/PursuitWeb/
8. SAIL Project. http://www.sail-project.eu/
9. Chai, W.K., Wang, N., et al.: CURLING: content-ubiquitous resolution and delivery infrastructure for next-generation services. IEEE Commun. Mag. **49**(3), 112–120 (2011)
10. Tyson, G., Sastry, N., Cuevas, R., Rimac, I., Mauthe, A.: A survey of mobility in information-centric networks. Commun. ACM **56**(12), 90–98 (2013)
11. Varvello, M., Rimac, I., Lee, U., Greenwald, L., Hilt, V.: On the design of content-centric MANETs. In: 2011 Eighth International Conference on Wireless On-Demand Network Systems and Services (WONS), pp. 1–8, January 2011
12. Oh, S.Y., Lau, D., Gerla, M.: Content centric networking in tactical and emergency MANETs. In: IFIP Wireless Days (WD), pp. 1–5 (2010)
13. Meisel, M., Pappas, V., Zhang, L.: Listen first, broadcast later: topology-agnostic forwarding under high dynamics. In: Annual Conference of International Technology Alliance in Network and Information Science, London, pp. 1–8 (2010)
14. Scott, K., Burleigh, S.: Bundle protocol specification. RFC 5050, November 2007. http://tools.ietf.org/html/rfc5050
15. Su, J., Scott, J., Hui, P., Crowcroft, J., de Lara, E., Diot, C., Goel, A., Lim, M.H., Upton, E.: Haggle: seamless networking for mobile applications. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 391–408. Springer, Heidelberg (2007)
16. Pitkänen, M., Karkkainen, T., Ott, J., Conti, M., Passarella, A., Giordano, S., Puccinelli, D., Legendre, F., Trifunovic, S., Hummel, K., May, M., Hegde, N., Spyropoulos, T.: SCAMPI: Service platform for soCial Aware Mobile and Pervasive computIng. In: Proceedings of Mobile Cloud Computing (MCC), Helsinki, Finland, August 2012
17. Tyson, G., Bigham, J., Bodanese, E.: Towards an information-centric delay-tolerant network. In: Proceedings of 2nd IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN), Torino, Italy, April 2013
18. Anastasiades, C., Uruqi, A., Braun, T.: Content discovery in opportunistic content-centric networks. In: Proceedings of 5th IEEE WASA-NGI, Clearwater, FL, pp. 1048–1056, October 2012
19. Batista, B., Mendes, P.: ICON - an information centric architecture for opportunistic networks, Poster. In: 2nd IEEE INFOCOM Workshop on Emerging Design Choices in Name-Oriented Networking (NOMEN), Italy, Torino, April 2013
20. Xylomenos, G., Vasilakos, X., Tsilopoulos, C., Siris, V.A., Polyzos, G.C.: Caching and mobility support in a publish-subscribe internet architecture. IEEE Commun. Mag. **50**(7), 52–58 (2012)
21. Vasilakos, X., Siris, V.A., Polyzos, G.C., Pomonis, M.: Proactive selective neighbor caching for enhancing mobility support in information-centric networks. In: Proceedings of ACM SIGCOMM Workshop on Information-Centric Networking (ICN), Helsinki, Finland, pp. 61–66, August 2012
22. Ravindran, R., Lo, S., Zhang, X., Wang, G.: Supporting seamless mobility in named data networking. In: Proceedings of IEEE FutureNet V, Ottawa, Canada, June 2012
23. Moskowitz, R., Nikander, P.: Host Identity Protocol (HIP) Architecture. IETF RFC 4423, May 2006

24. Hermans, F., Ngai, E., Gunningberg, P.: Global source mobility in the content-centric networking architecture. In: Proceedings of 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design - Architecture, Algorithms, and Applications (NoM), pp. 13–18, June 2012
25. CCNx. http://www.ccnx.org/
26. Anastasiades, C., Schmid, T., Weber, J., Braun, T.: Opportunistic content-centric data transmission during short network contacts. In: Proceedings of IEEE WCNC, Istanbul, Turkey, April 2014
27. Anastasiades, C., El Maudni, W., Braun, T.: Agent-based content retrieval for opportunistic content-centric networks. In: Proceedings of 12th WWIC, Paris, France, May 2014
28. Shen, S.-H., Gember, A., Anand, A., Akella, A.: Refactoring content overhearing to improve wireless performance. In: Proceedings of 17th ACM MobiCom, Las Vegas, USA, pp. 217–228, September 2011

http://www.springer.com/978-3-319-10833-9

Wireless Networking for Moving Objects
Protocols, Architectures, Tools, Services and Applications
Ganchev, I.; Curado, M.; Kassler, A. (Eds.)
2014, XVI, 301 p. 96 illus., Softcover
ISBN: 978-3-319-10833-9