Global Source Mobility in the Content-Centric Networking Architecture

Frederik Hermans
frederik.hermans@it.uu.se

Edith Ngai
edith.ngai@it.uu.se

Per Gunningberg
perg@it.uu.se

Department of Information Technology
Uppsala Universitet, Sweden

ABSTRACT

The Content-Centric Networking (CCN) architecture, a clean-slate network design, borrows its routing concepts from IP. If content is located on mobile sources, CCN also inherits some of the mobility problems known from IP. In this paper, we explore the design space of CCN mobility solutions by revisiting well-known IP approaches that aim to solve a remarkably similar problem. While mobility solutions may be quite similar in both architectures, we find that a locator/identifier split should be implemented at the network layer in CCN to prevent temporary, topology-dependent information to leak into content that ought to be permanent. Mobility handling further benefits from CCN's security model and multipath forwarding. To provide a starting point for further research, we present a simple mobility approach based on an explicit locator/identifier split.

1. INTRODUCTION

Mobile Internet access has become the norm rather than the exception, with an estimated 1.2 billion mobile broadband subscriptions for 2.4 billion Internet users in 2011 [8]. Hence, clean-slate information-centric network designs, which promise more efficient use of network resources by replacing the Internet's host-based communication with the retrieval and publication of named content, must support large-scale endpoint mobility.

We consider the problem of retrieving content from mobile sources in the CCN architecture. By mobile sources we mean nodes that serve content and that frequently change their topological location, such as smartphones switching between WiFi and 4G access networks. In CCN, content requests are routed towards sources by longest-prefix matching of hierarchical content names against forwarding tables [9, 17]. CCN does not implement a locator/identifier split, as content names are used for both routing of requests and identification of requested content. Thus, if a source changes its topological location, forwarding tables must be updated to ensure that requests reach the source at its new attachment point.

A similar problem exists in IP, where mobile nodes usually obtain a new IP address at each new attachment point, which breaks ongoing transport sessions and complicates access of services on mobile nodes. A range of mobility solutions have been proposed to cope with the problem. In this paper, we revisit some of the proposed IP mobility solutions and consider their applicability and ramifications in CCN, as the problems in both architectures are similar. While we do not claim that an IP-inspired solution will necessarily give the best way to handle mobility in CCN, we find it useful to look at mobility in CCN from an IP perspective to appreciate the similarities and differences of the problem. Our conclusion is that in contrast to IP, a CCN mobility solution must not depend on encapsulation or shim layer approaches, as these hide permanent content identifiers from routers and thereby thwart CCN's caching efforts. Furthermore, CCN's security model and multipath forwarding simplify the design of mobility solutions. We describe the design and implementation of a simple mobility approach based on our considerations.

In summary, this paper's contribution is two-fold: First, we explore the design space of global mobility solutions in CCN from an IP perspective and describe differences and commonalities. Second, we provide a starting point for further research by describing a concrete approach to global source mobility in CCN.

2. CCN ARCHITECTURE OVERVIEW

In CCN, content such as a picture is divided into a set of individually named, smaller *content objects*. Content object names are hierarchical, human-readable and can be arbitrarily long, e.g., /sprint/atlanta/alice/z.jpg/1. Each content object carries a cryptographic signature, which is computed over the name, the actual content, and some metadata.

All communication is consumer-initiated. A consumer retrieves an individual content object by sending an *interest* that specifies the name of the desired content object. When a router receives an interest and has a copy of the content object in its local cache, it sends back the copy and does not propagate the interest further. If the router has no copy, it looks up the next-hop neighbor(s) to forward the interest to by performing a longest-prefix

match of the name of the requested content against its forwarding table. Each table entry maps a name prefix, e.g., /sprint, to a set of next-hop neighbors. The interest is forwarded until it eventually reaches a node that has the requested content or can produce it. This node sends back the content object, which is propagated back to the consumer by following the reverse path of the interest. Routers that forward a content object store it in their local cache to directly answer later requests.

Forwarding tables are populated by prefix announcements. If a node has content under a certain name prefix, it announces this prefix to the network, and the announcement is propagated similar to BGP. We assume that names are provider-assigned [17], i.e., the top-level components of a name denote the provider of the content source. Provider-assigned names facilitate aggregation of prefixes, which in turn helps scalability via small forwarding tables.

Interests may be forwarded on multiple paths if a node has more than one potential path to the requested content, e.g., because the node is multi-homed, the source is multi-homed, or the content is available at multiple sources. Each node's *strategy layer* is responsible for path selection. Duplicate interests are subsumed at the network layer. A detailed description of CCN, including handling of mobile consumers (as opposed to mobile sources), can be found in [9, 17].

3. GLOBAL MOBILITY IN IP AND CCN

In IP, consider a mobile node (MN) that is at first connected to one provider's network, then disconnects, and connects to another provider's network[1]. Since the MN's IP address in general is not topologically correct at its new attachment point, it obtains a new address, thus breaking ongoing transport layer sessions that are bound to the old IP address (such as TCP sessions). Furthermore, a remote host cannot use the MN's old IP address any more to access services run on the MN.

Now consider an Internet-scale CCN network (Fig. 1) that is divided into a core and different provider networks that we call domains. The network's routers are configured to forward interests for the prefix /X to domain X, etc. A mobile source (MS) is usually connected to its home domain Z, and serves content under the prefix /Z/ms. The MS then disconnects from Z and connects to domain Y instead. A consumer (C) now cannot retrieve any content from MS that has not been cached in the network, because interests for /Z/ms will be routed to domain Z. Thus, the problem is very similar to the mobility problem in IP. Our goal is to begin an exploration of design options for mobility solutions that allow consumers to retrieve content from an MS regardless of its current attachment point.

[1]This is what we mean by *global* mobility: a node changing provider networks, thus making large jumps in the topology.

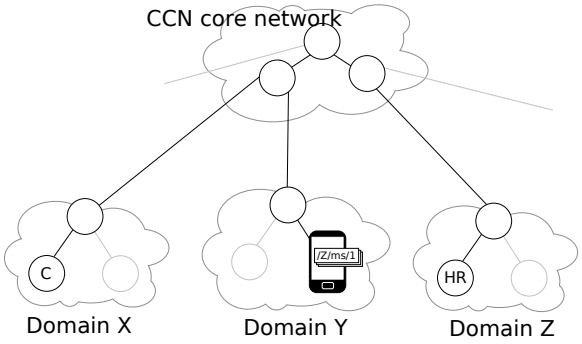

Figure 1: Three domains X, Y, and Z are interconnected through the CCN core. The mobile source, depicted as a smartphone, is usually connected to domain Z (its home domain), but is currently connected to Y.

3.1 Routing-based solutions

In the IP solution Connexion [3], mobile networks re-announce their reachability via BGP announcements from each new attachment point. Core routers update their forwarding tables in response to these announcements. The approach raised scalability concerns, since a large number of mobile entities would generate an overwhelming number of route updates and cause routing tables to grow due to prefix de-aggregation.

A CCN equivalent of Connexion would require mobile sources to re-announce their content prefixes from each new attachment point. The same scalability issues arise, especially when a source serves content for multiple distinct prefixes that cannot be aggregated. We conclude that global routing updates are inadequate to cope with source mobility.

3.2 Which layer to handle mobility?

In the Internet, mobility solutions can be found at the network, transport and application layer. Where should CCN handle source mobility?

A number of ad-hoc mobility solutions can be found in IP applications, but such approaches are unsuitable for CCN. CCN simplifies development of networked applications, because applications can be ignorant about where content is located. Handling mobility within applications, for example by renaming content after roaming events, would undo this strength.

SCTP handles IP mobility at the transport layer [15], HIP [12] and SHIM [13] introduce shim layers between network and transport to decouple the identifier and locator role of a destination address, and Mobile IP [10] and LISP [4, 5] use encapsulation at the network layer for the same purpose. These approaches all suffer from the same problem when applied to CCN: they require content objects—the network layer abstraction of data packets—to include topology-dependent information either in their names or in their payload. When

this topology-dependent information changes, the content objects change too. This is at odds with the idea of caching, which requires content objects to be permanently meaningful by themselves, independent of where the source happened to be when a content object was requested.

We can clarify the issue by an example using encapsulation. An MS has a content object named B, but can only receive interests for a prefix A at its current attachment point. A consumer wants to retrieve B, and via some mapping database learns that the source can only receive interests for A. The consumer sends an interest /A/#/B, where # is some agreed upon marker. This interest is routed to the MS. The MS cannot respond with the desired content B directly, because the name B does not match the interest name. (In CCN, a content object satisfies an interest only if the interest's name is a prefix of the content object's name.) Thus, MS encapsulates B in a content object named /A/#/B, which is propagated back to the consumer. All routers on the path between MS and the consumer now cache the wrapping content object with its temporary, location-dependent name, rather than B, the content object of interest. Furthermore, a router that sees an interest for B cannot respond with its cached content object /A/#/B, nor can it supply a cached copy for a request for /C/#/B. Encapsulation and shim layer approaches thwart CCN's caching efforts, because temporary information is leaked into content identifiers or content itself.

A further drawback is the necessity to re-sign content objects. Content objects in CCN must be signed, and the signature is computed over the content object's name, metadata and its payload. Thus, if a content object changes because it includes some temporary locator that changes, the signature must be recomputed. This places an undesirable computational burden especially on battery-powered devices.

In conclusion, mobility in CCN should be handled at the network layer in a way that stable content identifiers are exposed to the network's routers, and in a way such that content objects do not contain any topology-dependent, temporary information. We believe that a decoupling of the identifier and locator roles of a content name is necessary to reach this goal.

3.3 Separate namespaces for identifiers and locators?

If we separate content identifiers from content locators, the question arises what namespaces they should come from. IP mobility solutions vary in this respect: Mobile IP and LISP use IP addresses for both functions, whereas HIP introduces a new flat namespace of self-certifying names to represent host identities.

Regarding content identifiers, there is an ongoing debate about the merits of flat, self-certifying names vs. hierarchical, human-readable names [6, 14]. On behalf of hierarchical names, we would like to reinforce the point that hierarchical names help build scalable, simple lookup systems (such as DNS). Also, hierarchical names simplify generating content in response to a request by allowing to request content by a prefix of its name, rather than by its full name. However, the question of how identifiers are represented has architectural implications far beyond mobility handling, and hence, we do not aim to settle the issue here.

Regarding locators, we note that if IP routing principles are to be retained in CCN, locators must be hierarchical.

3.4 Which entities should handle mobility?

IP solutions aim to be compatible with legacy nodes, and are often designed to require upgrades in as few nodes as possible. Due to the lack of widespread deployment, compatibility is not an important issue in CCN, since upgrading all CCN nodes is still feasible. Thus, it is reasonable to assume an MS in CCN to participate in mobility handling, since each MS has the best knowledge about its own connectivity. By the same argument, consumers may be involved in mobility handling.

CCN mobility solutions may differ in regard to what functionality they require from other network elements, which could for example supply cached content in case of disconnection of an MS. A comparison of concrete proposals for source mobility will help to understand the implications of different approaches.

3.5 Mobility anchors or explicit identifier resolution?

A design option related to the previous section is whether a CCN mobility solution should use mobility anchors like the home agent in Mobile IP, or whether an explicit resolution step should be used to resolve identifiers to locators, like in LISP or HIP. Again, we believe that study of concrete mobility proposals for CCN will be necessary to settle this matter.

Requiring explicit resolution together with a strict separation of identifiers and locators may help the scalability of the CCN routing infrastructure in general. This goal is also pursued by LISP and ILNP [2] in IP. The idea is that core routers only store topological locators in their routing table, instead of content prefixes which may reflect organizational hierarchies rather than topology. This would help to keep routing tables small in case of multi-homing and mobility.

If, however, mobility is a rare case, the use of mobility agents allows to avoid the lookup overhead for content from static sources, at the cost of a potential detour for retrieving content from mobile sources.

3.6 How to provide session continuity?

A mobility solution should minimize delays in ongoing communication due to a roaming event of a mobile node, especially when unanticipated. In IP, this is typically handled by the mobile node explicitly notifying correspondent nodes after reconnecting at a new attachment point. In CCN, a source cannot explicitly notify consumers of its movement, because consumers are anonymous—an interest carries no information about which node(s) sent it. As a work-around, consumers could be required to register a notification prefix with the source, though this increases complexity.

CCN's support for multi-path interest forwarding alleviates the hand-over problem. If a consumer has multiple paths to a source, but does not know which of them are alive due to source mobility, it can send its interest on all paths. Only the interest to arrive first is received by the MS's application, as later duplicate interests are automatically subsumed. While in IP a host may also send a packet along multiple paths, duplicates are delivered to the mobile node if at least two paths are alive, causing adverse effects on TCP connections.

3.7 Security aspects

Part of the complexity of IP mobility solutions stems from the necessity to authenticate bindings between identifiers and locators. This problem may be easier in CCN, if bindings are stored as regular content objects. In that case, CCN already allows any node to verify the integrity and authenticity of a binding. However, it must then be possible to obtain the key with which a binding is signed without requiring trust in the binding itself.

3.8 Deployment

If the mobility solutions in IP did not reach widespread deployment yet, why should an IP-inspired mobility solution for CCN not face the same acceptance problems? The lack of deployment can partly be attributed to the operators' unwillingness to upgrade existing systems, rather than to technical deficits in the proposals. We believe that a sufficiently flexible mobility solution will not face the same acceptance issues, if it is developed well before large-scale CCN deployment.

4. LOCATOR/IDENTIFIER SPLIT IN CCN FOR SOURCE MOBILITY SUPPORT

We now describe how a locator/identifier split can be implemented in CCN, and how it allows to implement a mobility solution that does not require global routing updates and ensures that content objects are permanent. Similar to Mobile IP, our approach uses a mobility anchor, but it does not rely on encapsulation.

We assume the scenario and topology depicted in Fig. 1. The mobility agent, which we refer to as the *home repository* (denoted HR in the figure), is a node in the mobile source's home domain that does not move and that can receive interests on behalf of the mobile source. A consumer (C) wants to retrieve content from the mobile source (MS).

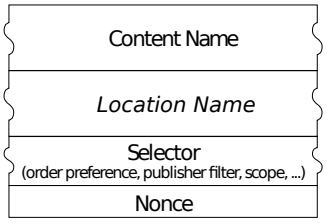

Figure 2: The interest message format has been modified to include an optional location name. (Adapted from [9])

4.1 Locator/ID split implementation

We modify the format of interest messages to contain an optional *location name* field, which may be specified in addition to the mandatory content name. The modified interest message format is shown in Fig. 2. When a router receives an interest that specifies a location name, it will first match the interest's content name against its cache to check whether it has a matching copy of the requested content. If it does not have a copy, the router must propagate the interest; to this end, it matches the *location name* against its forwarding table to determine the next hop neighbors the interest needs to be forwarded to. Thus, an interest that specifies a location name requires two lookups at a caching router. Interests that do not specify a location name are propagated on the content name, as in the original design.

A location name is a regular CCN name, but it does not refer to content. Instead, it is merely used as a routing locator and conceptually refers to the set of sources that can receive interests that are routed on the location name. It may appear counter-intuitive to introduce location names to CCN. Note, however, that this location info is only used by the network layer to retrieve content (which is its primary purpose), and it is not exposed to applications.

4.2 Binding content prefixes to locations

When the MS attaches to a domain other than its home domain, it is assigned some prefix that it can receive interests for. E.g., it could be assigned /Y/guest12 in the example from the previous section. This prefix will be used as the source's location name. The MS then publishes a *binding info* that contains (a) the prefix of content located at the MS, and (b) the MS's location name[2]. A binding info is an ordinary content object,

[2]If the MS is multi-homed, it may state multiple location

and therefore inherits the security model of CCN, so any node can ascertain its authenticity. The MS publishes the binding info at the home repository, from where it can be retrieved by any node in the network. Whenever the MS changes its location, it updates the binding info. This update is reflected by increasing a version component in the name of the binding info content object.

4.3 Retrieving content from mobile sources

Routing interests via the home repository: The HR, upon receiving a binding info, learns about the MS's content prefix and its location name. When the consumer sends an interest I for content from the MS, the interest will be routed to the MS's home domain, where the HR receives it. In response, the HR sends out a new interests I' that is identical to I, except that I' specifies the new location name field, which it sets to the MS's location name. The new interest I' will be routed towards the MS. The MS responds with the requested content object, which will be propagated back to the consumer via the HR. This is shown in Fig. 3. Routers that forward the content object cache it under the persistent content name.

Routing interests via the HR enables consumers to retrieve content from the MS regardless of its attachment point, but interests and content objects may take a detour if the HR is not on the path between consumer and MS. We now consider how to route interests directly to the MS.

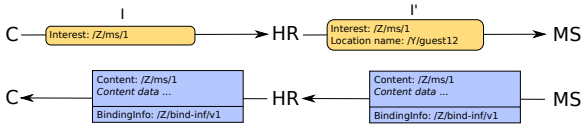

Figure 3: C retrieves content from MS via HR

Routing interests directly to the mobile source: To send interests directly to the MS, the consumer must obtain the binding info, which specifies the MS's location name. The MS piggy-backs the name of the binding info on each outgoing content object. When the consumer receives the first content object from the MS—indirectly via the HR—it thus also receives the name of the binding info. It retrieves the binding info, verifies it, and extracts the content prefix and location name from it. The consumer sets the location name field for all subsequent interests for content from the MS. Thus, the interests are routed directly to the MS without taking a detour via the HR, as shown in Fig. 4. The content objects take the direct path, too, and routers cache them under their persistent content names.

All this functionality is implemented at the network layer, which takes responsibility for publishing and retrieving binding infos, as well as setting location names for outgoing interests. Mobility and locations are transparent to the applications on the MS and consumer, which merely have to care about providing and requesting named content, respectively.

names; for the sake of clarity, we assume the MS to be single-homed for the rest of the description.

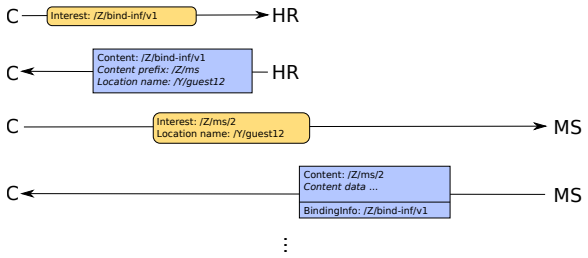

Figure 4: C learns the location name and retrieves content directly from MS

Handling location changes: If the MS changes its attachment point again, its location name becomes invalid and interests from the consumer and HR will no longer reach the source. As soon as it is assigned a new location name at the new attachment point, the MS updates the binding info and notifies the HR. Consumers exploit CCN's multipath forwarding to handle hand-overs. A consumer has two paths to content from the MS. On one path, interests are routed via the HR, and on the other path, interests are routed directly to the MS. If the interests that are routed directly are not satisfied any more because the location name has become invalid, the consumer's strategy layer will automatically fail over to the other path, and hence send the interests to the HR. The steps described previously are repeated, and the consumer learns about the new location name. In this way, unexpected source location changes can be handled transparently with moderate interruptions.

4.4 Implementation

We have implemented the described approach as a modification to the CCN prototype[3]. Basic experiments confirmed that the implementation correctly handles source mobility including unanticipated roaming events.

We take the amount of changes that were required as an indication that the approach is not in strong violation of the original CCN design: About 4% ($\sim$ 330 lines) of the code of the CCN daemon, and 1.4% ($\sim$ 180 lines) of the CCN library were changed. The additional software required for mobile sources, home repository, and consumers weighs in at about 1000 lines of code in total.

5. RELATED WORK

[3]Our implementation does not fully handle response-time prediction yet. However, we do not foresee any major problems in adding this functionality.

Meisel describes BOND [11], an adaption of CCN principles for use in mostly unstructured ad-hoc and delay tolerant networks. BOND does not address Internet-scale deployments. Wang et al. study the applicability of CCN in vehicle-to-vehicle communication [16]. We have described an earlier version of our approach in [7].

A survey of IP mobility can be found in RFC 6301[18].

Ahlgren et al. survey various information-centric network designs and also address changes of content location [1]. To summarize, most other designs allow an explicit name resolution step in which content names are resolved to locations. The elements involved in this name resolution need to be updated if a content source has moved in the topology.

6. CONCLUSION AND FUTURE WORK

In this paper, we have considered the design space for global mobility solutions in the CCN architecture from an IP perspective. Due to the similarity in routing and the lack of an identifier/locator split in both architectures, the problems are very similar, and IP mobility solutions are applicable to CCN if care is taken to ensure that content objects are permanent. We have described how a locator/identifier split can be implemented in CCN by adding a new field to interest messages, and how a mobility solution can be built on top of this modification.

We do not claim that an IP-inspired mobility solution is necessarily the best way to handle mobility in CCN, and we are open to more radical approaches. However, if CCN retains the IP routing approach, other mobility solutions are likely to contain elements that we know from today's IP mobility solutions. We thus hope for our work to be useful for other researchers.

In the future, we plan to develop a scalable mapping system for CCN that allows to map identifier prefixes to locators. This would allow to implement a crisper separation between locators and identifiers, and would make the use of a mobility agent unnecessary at the cost of an additional resolution step at the consumer. We also plan to evaluate the mobility solution presented in this paper.

7. REFERENCES

[1] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. A Survey of Information-Centric Networking (Draft). In *Information-Centric Networking*, Dagstuhl Seminar Proceedings, Feb. 2011.

[2] R. Atkinson. Work in progress: ILNP Concept of Operations. `http://tools.ietf.org/id/draft-rja-ilnp-intro-11.txt`, July 2011.

[3] A. L. Dul. Global IP Network Mobility using Border Gateway Protocol (BGP). Technical report, Boeing, Mar. 2006.

[4] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. Work in progress: Locator/ID Separation Protocol (LISP), version 12. `http://tools.ietf.org/html/draft-farinacci-lisp-12`, Mar. 2009.

[5] D. Farinacci, D. Lewis, D. Meyer, and C. White. Work in progress: LISP Mobile Node. `http://tools.ietf.org/html/draft-meyer-lisp-mn-06`, Oct. 2011.

[6] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker. Naming in content-oriented architectures. In *Proc. of the ACM SIGCOMM ICN Workshop*. ACM, 2011.

[7] F. Hermans, E. Ngai, and P. Gunningberg. Mobile Sources in an Information-Centric Network with Hierarchical Names: An Indirection Approach. In *Proc. of the 7th Swedish National Computer Networking Workshop*, June 2011.

[8] International Telecommunication Union. Key global telecom indicators for the world telecommunication service sector. `http://www.itu.int/ITU-D/ict/statistics/at_glance/KeyTelecom.html`, Nov. 2011.

[9] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *Proc. of CoNEXT*, pages 1–12, Dec. 2009.

[10] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard), June 2004.

[11] M. Meisel. *BOND: Unifying Mobile Networks with Named Data* . PhD thesis, University of California Los Angeles, USA, 2011.

[12] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson. Host Identity Protocol. RFC 5201, Apr. 2008.

[13] E. Nordmark and M. Bagnulo. Shim6: Level 3 Multihoming Shim Protocol for IPv6. RFC 5533, June 2009.

[14] D. K. Smetters and V. Jacobson. Securing network content. Technical report, PARC, Oct. 2009.

[15] R. R. Stewart. Stream Control Transmission Protocol. RFC 4960, Sept. 2007.

[16] L. Wang, R. Wakikawa, R. Kuntz, R. Vuyyuru, and L. Zhang. Data naming in vehicle-to-vehicle communications. In *Procs. of the 1st Workshop on Emerging Design Choices in Name-Orientied Networking*, NOMEN '12, Mar. 2012.

[17] L. Zhang, et al. Named Data Networking (NDN) Project. Technical report, PARC, Oct. 2010.

[18] Z. Zhu, R. Wakikawa, and L. Zhang. RFC 6301: A Survey of Mobility Support in the Internet, July 2011.