Kite: A Mobility Support Scheme for NDN

Yu Zhang Hongli Zhang
Harbin Institute of Technology
{yuzhang, zhanghongli}@hit.edu.cn

Lixia Zhang
UCLA
lixia@cs.ucla.edu

ABSTRACT

Named Data Networking (NDN) natively supports the mobility of data consumers through its data-centric design and stateful forwarding plane. However, the mobility support for data producers remains open in the original proposal. In this paper, we introduce Kite, a design of mobility support for NDN. Kite leverages the state of the Pending Interest Table (PIT) at each router to reach mobile nodes. We describe how Kite can support typical scenarios including group communications among mobiles.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: COMPUTER-COMMUNICATION NETWORKS; C.2.1 [**Network Architecture and Design**]: Packet-switching networks—*Internet*

Keywords

Mobility Support; Named Data Networking

1. INTRODUCTION

Named Data Networking (NDN, aka CCN) [3, 6] has two built-in features that are beneficial to mobility: its *data-centric* design and *stateful* forwarding plane. First, every packet, either Interest or Data, carriers a data name only, instead of any address. Second, the states of data requests, namely Interests packets in the Pending Interest Tables (PITs) of the forwarding plane, enable the reverse path forwarding of Data packets, which makes the location of data consumers transparent to the routing plane and data producers. Thus, NDN natively supports data delivery to mobile consumers, as described in the seminal paper of NDN [3]. However, the paper did not sketch out any specific design for supporting mobile producers.

Several recent efforts have focused on filling in this missing piece. CBIS [2] introduced custodians as intermediaries between names and endpoints. Both [1] and [5] proposed to let each mobile producer keep its location updated at a home agent. [4] lets a mobile node (MN) issue a special Interest to its previous location to set up a reverse path by creating FIB entries.

In this paper we present *Kite*, a mobility support scheme for NDN. The key idea of Kite is to fully exploit NDN's forwarding states in PITs (a hop-by-hop *trace*) to keep track of an Interest issuer on the move. This idea is analogous to flying a kite: a kite (like an application) will be reachable along a string (like a trace) from hands (like an anchor). A full description of the design is in an NDN technical report[7].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).
ICN'14, September 24–26, 2014, Paris, France.
ACM 978-1-4503-3206-4/14/09.
http://dx.doi.org/10.1145/2660129.2660159 .

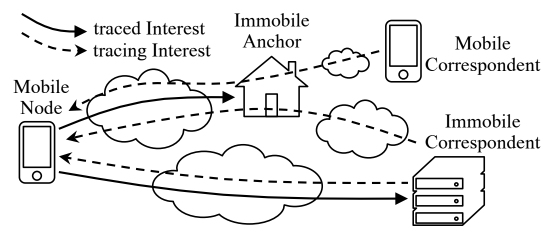

Figure 1: Framework of Kite

2. KITE

NDN forwards each Data following the trace left by an Interest. Kite allows each specially marked Interest to be forwarded in the same way, along the trace left by another Interest. We call an Interest which leaves a trace back to an MN a *traced Interest*, and an Interest that traverses the trace left by a traced Interest *tracing Interest*. As shown in Figure 1, a trace may be set up directly between an MN and a correspondent node (CN) or indirectly via an immobile anchor.

We introduce a new forwarding mechanism, *Interest trace forwarding*, into NDN's forwarding plane. For each *tracing Interest* I_{tg}, we add (1) a new `TraceName` field to indicate which traced Interest I_{td} to be traced, and (2) a new `TraceOnly` flag to indicate how to forward the tracing Interest: If the flag is unset, I_{tg} will be forwarded along both the routes in FIBs and trace in PITs; Otherwise I_{tg} will only follow the trace of I_{td} if a match of its trace name is found in the PIT. I_{tg} will be sent (fan out) via the incoming interfaces of matched I_{td} entry.

3. KITE PROTOCOL DESIGN

Generally speaking, we may categorize mobile applications into the following 4 scenarios. (1) An MN, e.g., a smart

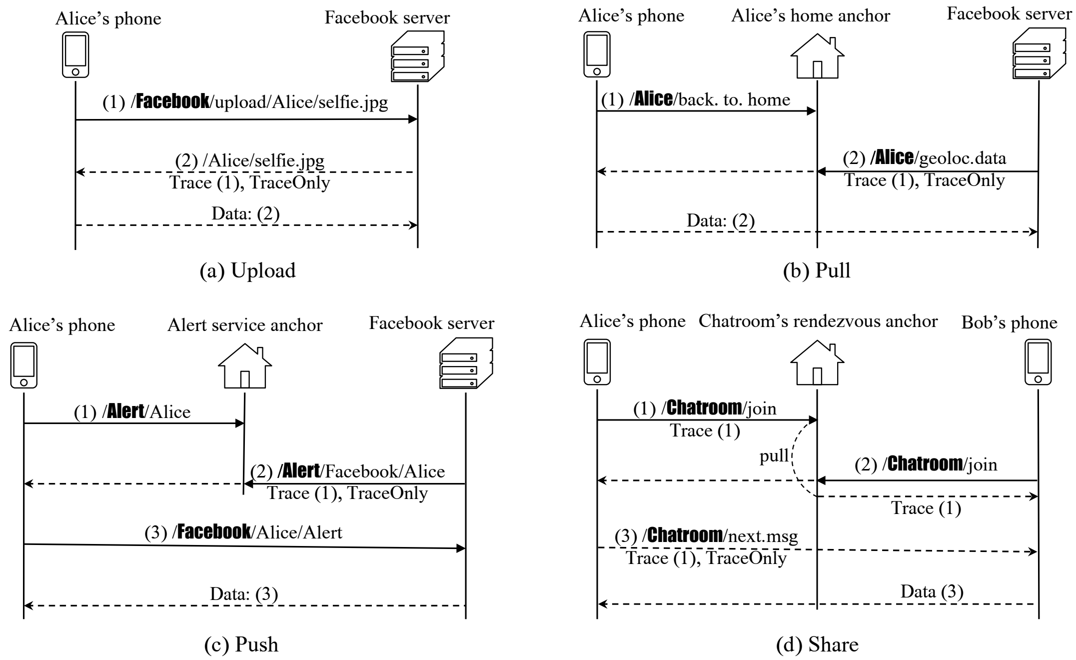

Figure 2: Message exchanges in mobile application protocols. Bold prefixes are in FIBs.

phone, *uploads* data to a stationary CN, e.g., a server; (2) a stationary CN *pulls* data from a mobile; (3) a stationary CN *pushes* data to a mobile; and (4) a group of mobiles *share* data among each other, such as chat via phones directly without a central server. In this case both producers and consumers are mobiles.

The upload scenario can be directly supported by Kite. For example, when Alice uploads her selfie from her phone to a Facebook server, an stationary consumer, the packet exchange is shown in Figure 2 (a). First, the phone sends Interest-1 to both notify the server of the data name to be uploaded, and set up a trace back to the mobile. The server then fetches the data with a tracing Interest-2, whose `TraceName` is derived from Interest-1. In the case of move-before-get, i.e., Alice moves before receiving Interest-2, the phone resends Interest-1, which will pull Interest-2 back to the phone.

Kite can support data pull and push scenarios by utilizing stationary anchors as shown in Figure 2 (b) and (c), respectively. An MN sends traced Interests which follow routes in FIBs to reach its anchor. Tracing Interests from stationary CNs first follow routes in FIBs to reach the anchors, then traverse the traces in PITs to MNs. With the `TraceOnly` flag set, the tracing Interest is only forwarded along traces after arriving at the anchor.

In the share scenario, Kite helps group members build a *bidirectional (shared) tree* rooted at a stationary anchor. As shown in Figure 2 (d), Alice and Bob each need to first send a join Interests-1 and -2, which meet at the anchor. Interest-2 will then trace Interest-1 to reach Alice, and Interest-1 reaches Bob. As a result, a bidirectional trace between Alice and Bob is set up. Afterward, Alice and Bob can express Interests to each other on the tree.

As a proof-of-concept of Kite, we implemented the protocols for the upload and share scenarios in ndnSIM and conducted a preliminary evaluation[7].

4. CONCLUSION

Kite contributes two new features to the mobility support of NDN. 1) *Locator-free*: There is no explicit locator for MNs which are implicitly addressed by the hop-by-hop states instead. 2) *Scenario-aware*: As traces are generated and utilized directly by application protocols, protocol developers are partially empowered to devise their own designs of mobility support tailored to their scenarios.

5. ACKNOWLEDGMENTS

This work is supported by National NSF of China (No. 61202457) and State Scholarship Fund from China Scholarship Council (No. 201206125041).

6. REFERENCES

[1] F. Hermans, E. Ngai, and P. Gunningberg. Global source mobility in the content-centric networking architecture. In *NoM '12*, 2012.

[2] V. Jacobson et al. Custodian-based information sharing. *IEEE Communications Magazine*, 50(7):38–43, 2012.

[3] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Networking named content. In *CoNEXT '09*, 2009.

[4] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom. Mobility support in content centric networks. In *ICN '12*, 2012.

[5] J. Lee, S. Cho, and D. Kim. Device mobility management in content-centric networking. *IEEE Commun. Magazine*, 50(12):28–34, 2012.

[6] L. Zhang et al. Named data networking. *Tech. Rep. NDN-0019*, 2014.

[7] Y. Zhang, H. Zhang, and L. Zhang. Kite: A mobility support scheme for ndn. *Tech. Rep. NDN-0020*, 2014.