KITE: Producer Mobility Support in Named Data Networking

Yu Zhang, Zhongda Xia
{yuzhang,xiazhongda}@hit.edu.cn
Harbin Institute of Technology

Spyridon Mastorakis, Lixia Zhang
{mastorakis,lixia}@cs.ucla.edu
UCLA

ABSTRACT

In Named Data Networking (NDN), mobility of data consumers is natively supported by the stateful forwarding plane. However, additional mechanisms are needed, so that requests for data can be forwarded toward a mobile data producer. In this paper, we present KITE, a trace-based producer mobility support that further exploits the stateful forwarding plane of NDN. KITE takes a soft-state approach to create a hop-by-hop path between a reachable rendezvous server and a mobile producer through authenticated Interest-Data exchanges. KITE is locator-free, transparent to data retrieval and routing, and abuse-proof. We show how KITE supports various mobile communication scenarios, and name-based rendezvous at the network layer. A KITE prototype is implemented and evaluated.

CCS CONCEPTS

• **Networks → Network mobility**; *Network design principles*; *Network protocol design*;

KEYWORDS

Named Data Networking (NDN), Producer Mobility, Name-Based Rendezvous

ACM Reference Format:
Yu Zhang, Zhongda Xia and Spyridon Mastorakis, Lixia Zhang. 2018. KITE: Producer Mobility Support in Named Data Networking. In *ICN '18: 5th ACM Conference on Information-Centric Networking (ICN '18), September 21–23, 2018, Boston, MA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3267955.3267959

1 INTRODUCTION

Named Data Networking (NDN) [30] is a proposed Internet architecture, featuring a data-centric and consumer-driven communication model. Mobility of nodes requesting data (consumers) is natively accommodated by NDN's stateful forwarding plane. However, additional mechanisms are needed to forward requests for data that is produced by a Mobile Producer (MP). This is known as the *producer mobility* problem in NDN. Although a number of solutions to support producer mobility have been proposed (§ 2), those proposals are not quite as "simple" as NDN's native support for consumer mobility. Our motivation is straightforward: can we support producer mobility in a "relatively simple" way? In other words, to what extent can we further exploit the native NDN features to support producer mobility?

This paper introduces *KITE*, a tracing-based producer mobility support solution in NDN (§ 3). With the help of an immobile rendezvous server (RV) that is reachable through the routing plane, an MP can be reachable along a "breadcrumb trail", namely *trace*, which is built by an authenticated Interest-Data exchange between the MP (as a consumer) and the RV (as a producer). This is analogous to flying a kite: a kite (MP) is attached along a string (trace) to a hand (RV).

To achieve simplicity, KITE shares two major similarities with NDN's native support for consumer mobility.

- **Prerequisite of a correspondent with a routable prefix:** For a mobile consumer, a producer is its correspondent; in KITE, an RV serves as the correspondent of the MP.
- **Locator-freeness:** A mobile node is reachable along hop-by-hop soft-state forwarding paths from its correspondent, without any explicit locator.

Thanks to the above similarities, KITE possesses two features:

- **Transparency to data retrieval:** KITE operates at the network layer, and does not change the data retrieval process.
- **Transparency to routing:** KITE makes producer mobility transparent to the routing plane. The forwarding information set by KITE does not override nor is overridden by routing.

Incorporating security considerations from scratch, KITE attains:

- **Abuse-proof by design:** KITE protects trace establishment by authenticating the Interest-Data exchange process.

The reachability of the MP can be guaranteed in KITE (§ 4). KITE provides general network layer support for various mobile application scenarios (§ 5). A prototype of KITE has been implemented and evaluated. (§ 6). Architectural implications and security considerations of KITE are also discussed (§ 7).

2 RELATED WORK

2.1 NDN Mobility

While NDN supports 1) consumer mobility natively, producer mobility solutions fall into two major categories [32]: MP chasing, and data rendezvous; the former delivers Interests to the MP by keeping track of the MP's movement, while the latter makes the MP's data available at a named location. MP chasing solutions can be further categorized into: 2) mapping-based, 3) tracing-based, and 4) routing-based solutions. 5) Data depot is a representative type of data rendezvous solutions.

1) Consumer Mobility: Despite their similarities on routable correspondents and locator-freeness, the native consumer mobility support and KITE mainly differ in three aspects: 1) In KITE, RVs serve as indirection between consumers and producers; 2) a consumer uses one Interest to set up the forwarding path in PITs to retrieve

ICN '18, September 21–23, 2018, Boston, MA, USA

ACM ISBN 978-1-4503-5959-7/18/09…$15.00
https://doi.org/10.1145/3267955.3267959

one Data back; in KITE, an MP uses an Interest-Data exchange to set up the forwarding path in FIBs for consumers' Interests with a specific prefix; 3) a consumer sets up trace only when requesting data; in KITE, an MP should set up trace proactively or on demand in anticipation of data retrieval.

2) Mapping-Based Solutions: Those solutions require the MP to be reached by an explicit locator which generally is a routable prefix of the current *Point of Attachment (PoA)*. The MP updates the mapping between its own data name prefix and its current routable prefix in a mapping server. The server may provide the mapping information to consumers [1], or act as a proxy to fetch data from the MP for the consumers [4, 15], or be a hybrid of the two [10, 13, 16, 29]. KITE is locator-free and only requires RVs to possess routable prefixes.

3) Tracing-Based Solutions: In [9, 34], a hop-by-hop "breadcrumb trail", namely trace, is built and updated between the MP and an immobile RV. The RV makes routing announcements for the MP's data name prefix. Consumer Interests are forwarded toward the RV first, and then follow the trace to the MP. Zhang et al. [34] proposed the initial idea of KITE. In this paper, we present a new design that: 1) uses authenticated Interest-Data exchanges to trace MPs; 2) leverages naming conventions to separate trace setup and data retrieval and to avoid changing the packet format; and 3) stores the forwarding information in FIB rather than in PIT, simplifying the forwarding process.

4) Routing-Based Solutions: The key challenge to keeping track of the MP through the routing plane resides in the scalability of routing updates due to mobility. Several solutions [3, 13, 22] *temporarily* restrict the scope of routing updates to the path from the current PoA to the previous PoA. That scope-restricted update mechanism looks like tracing-based solutions without immobile RVs.

However, additional assumptions on routing protocols and mechanisms dealing with interferences between scope-restricted updates and global routing updates are needed to guarantee the reachability of MPs. Furthermore, the reachability information on the MP's whereabouts has to be eventually updated throughout the routing plane. For example, in link-state routing protocols, such as IS-IS and NLSR, mobility leads to changes on *reachability* information in the link-state database, i.e., which destinations can be reached directly via a given router or PoA. This is still confronted with routing scalability issues: 1) mobility may trigger global updates on reachability information and global routing recalculations; 2) each MP requires its own topologically-independent, thus difficult-to-aggregate, device-specific prefix in both RIBs and FIBs.

KITE's routing-transparency prevents mobility from triggering any global routing update and interfering with FIB entries updated through the routing plane. In KITE, it is not necessary for MPs to be reachable through a device-specific routable prefix. Multiple MPs can share the same RV and the same routable prefix. Routable prefixes announced by immobile RVs can be topology-dependent.

5) Data depot: The approach is to move the data produced by an MP to an easily accessible (e.g., immobile) location [32]. For example, CBIS [11] introduces data custodians, which are designated sources for data under specific prefixes. The mobility of custodians is supported by updating a Custodian-to-Endpoint table (CET), which lists the communication endpoints that can be used to reach each custodian. CBIS and KITE can benefit from each other: immobile custodians can serve as RVs in KITE; and KITE can reduce CET mapping updates for mobile custodians by maintaining forwarding paths to custodians.

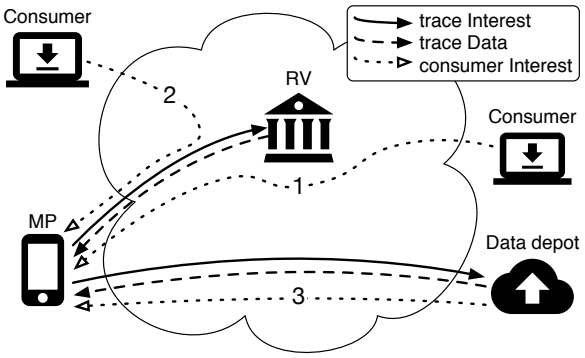


Figure 1: Conceptual Illustration of KITE

2.2 ICN Mobility

Tyson et al. [27] surveyed mobility support in various ICN designs. We stress that in most designs for future Internet architecture (including those not covered by the survey, such as XIA [2], and MobilityFirst [23]), data consumers are addressed by explicit locators or equivalents (e.g., source routing in LIPSIN [12]), while in NDN consumers are anonymously addressed by hop-by-hop states. Consequently, in NDN the mobility of consumers is transparent to producers and forwarders. KITE just extends the locator-freeness feature further to the mobility support of producers.

2.3 IP Mobility

The long and fruitful progress of mobility support research in IP context has been reviewed in [14, 38]. The RVs in KITE are similar to home agents in Mobile IP [20], but RVs do not change packets, and may be bypassed in the case of path shortcut. Cellular IP [28], HAWAII [21] and TIMIP [8] share the similar idea with KITE to set up hop-by-hop reverse paths back to mobile nodes, but they were designed for IP and requires major architectural changes, which limits them to only support micro-mobility. On the contrary, KITE leverages the native features of NDN. MSM-IP [18] pointed out that supporting multicast is kin to supporting mobility, so it is not surprising that KITE's soft relocation is almost identical to multicast. PIM-SM [7] and KITE are similar in various ways, including: 1) using soft states to deal with dynamics; 2) employing a fixed rendezvous point; 3) using existing routing information to explicitly build forwarding paths. But since NDN's stateful forwarding can naturally ensure loop-free forwarding, it is not necessary for KITE to do designated forwarder election or check reverse path forwarding like PIM-SM.

Overall, the combination of data-centricity, stateful forwarding, locator-freeness and transparency to data-retrieval/routing distinguishes KITE from those explicitly using separate locators and/or global routing/mapping.

3 THE KITE DESIGN

3.1 Overview

KITE is a tracing-based solution that operates in the forwarding plane by setting up a forwarding path from an immobile RV to an MP. Figure 1 shows the conceptual illustration of KITE. For the sake of simplicity, we assume that the MP's data names are covered by one routing prefix (§ 3.2). The RV announces a routing prefix in the routing plane, and the MP and the RV establishes trust with each other through exchanging public keys (§ 3.3). The MP issues a signed *Trace Interest* (TI) with a special ""`/trace`"" tag in the name to the RV. The RV verifies the TI and responds with a signed *Trace Data* (TD), which then travels back to the MP along the same path traversed by the TI. Upon receiving a TD, intermediate forwarders create or update FIB entries for the MP's data prefix, namely *trace*. The trace to the MP is constructed in a hop-by-hop fashion by saving the incoming faces of TI as next-hops. This process is called *trace setup* (§ 3.4). Each time the MP relocates, it starts the Interest-Data exchange again to build a new trace. The trace is soft-state, and will be purged if not refreshed. Therefore, the MP needs to actively keep the trace alive (§ 3.5). Consumers' Interests are forwarded towards the RV, meeting the trace to the MP either at the RV (consumer Interest 1) or half-way (consumer Interest 2) (§ 3.6), and then forwarded along the trace to the MP. Alternatively, a data depot may play the role of an RV, enabling MPs to directly upload data to it (consumer Interest 3) (§ 5.2).

The functions of four types of entities are summarized below:

- **Consumer** functions as a normal consumer.
- **MP** 1) conducts the TI/TD exchanges as a consumer, and 2) responds to consumers' Interests with Data as a producer.
- **RV** 1) makes routing announcements for routing prefixes, 2) verifies TIs and responds with TDs as a producer, and 3) forwards consumers' Interests that did not meet the trace halfway.
- **Forwarder** (besides functioning as an NDN forwarder) identifies TDs with the "trace" tag in names and manipulates the FIB accordingly for trace setup and maintenance.

Note that there is no one-to-one binding between any two of MPs, RVs and routing prefixes. An MP can produce data under multiple routing prefixes, and receive rendezvous services provided by application service providers, e.g., Facebook, organizations, e.g., UCLA, Internet service providers, e.g., ATT, or personal servers at home or office.

3.2 Namespace Design

The namespace design is illustrated in Figure 2. A *data name* is in the form of ""`/<routing prefix>/<tracing segment>/<data name suffix>`"". The *routing prefix* is announced to the routing plane by the RV. The *tracing prefix* is in the form of ""`/<routing prefix>/<tracing segment>`""; it is the MP's data prefix, and corresponding FIB entries that construct the trace are created in FIBs by the TI/TD exchanges. The *TI/TD prefix* is the tracing prefix with a ""`/trace`"" tag between the routing prefix and the tracing segment. The *TI/TD name* is the TI/TD prefix with additional *verification information* name components, including a timestamp, a nonce, some signature information, and a signature value. The RV uses the verification information to verify the authenticity of a TI.

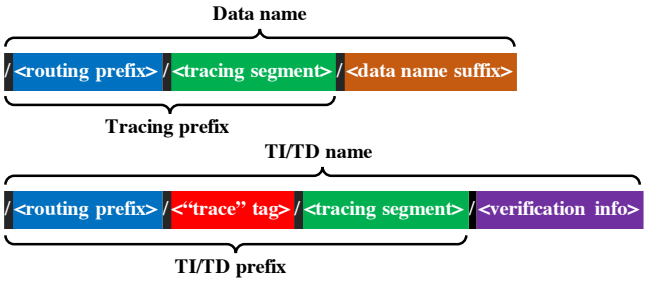


Figure 2: The namespace design of KITE

The namespace design serves the following purposes:

- The tracing prefix and the data name share a common prefix, i.e., the routing prefix, so that the trace and consumer Interests can meet at the RV.
- The tracing prefix is longer than the routing prefix, which separates the trace setup from routing plane. And according to longest prefix match, a consumer Interest will be forwarded to the MP along traces if present, otherwise forwarded toward the RV.
- The tracing prefix can be derived from the TI/TD prefix simply by removing the tag, while the two prefixes are different so that the trace setup is separated from the data retrieval process.

The specific naming design depends on application scenarios, as demonstrated in § 5; the routing prefix announced by the RV could be under a mobile device name, a user's name, an institutional name, or a service name. We do not assume that each MP has its own routing prefix, which in turn needs to be announced by the RV to the routing plane.

3.3 Preliminaries For Secure Trace Setup

To secure trace setup, three preliminary steps are required: routing announcement, trust establishment and timestamp synchronization. The security considerations are discussed later in § 7.2.

Routing announcement: The routing prefix is announced by the RV. We assume that the routing plane and all forwarders are trusted, so that Interests with the routing prefix can reach the RV and can not be hijacked.

Trust establishment: The MP and the RV learn about each other's public keys through offline contact or some public key distribution systems. Two public keys (and the corresponding private keys) held by the MP include 1) TI signing key, and 2) data signing key. Two public keys held by the RV include 1) TD signing key, and 2) timestamp signing key. The usage of those keys, such as the binding between keys and prefixes, are exchanged securely as well[1].

Timestamp synchronization: To resist replay attacks on trace setup, the MP and the RV may synchronize a clock for generating timestamps as a part of TI verification information. The synchronization process is in the form of regular data retrieval, i.e., the MP, as a consumer, fetches the RV's timestamp (*RV-TS*) from the RV: 1) the MP sends an Interest for *RV-TS* to the RV; 2) the RV responds with a Data message that contains the current *RV-TS* ; 3) after receiving this Data message, the MP sets its clock for generating TIs to $RV\text{-}TS + RTT/2$, where RTT is the observed delay of

[1]For a summary of the overall NDN security support, readers are referred to [36].

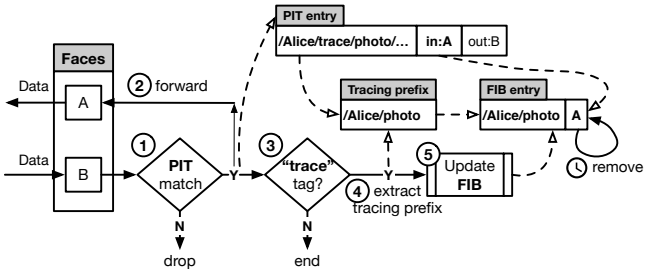


Figure 3: Trace setup: processing of a TD

this data retrieval. Note that that synchronized clock for the MP is not necessarily its system clock, and the freshness of a timestamp should be judged with a reasonable grace period.

3.4 Trace Setup

The trace setup process relies on an authenticated Interest-Data exchange between the MP and the RV. Once a signed TI reaches the RV, it will be verified. If the verification fails, the RV can either ignore the TI or trigger a network-layer Negative ACKnowledgment (NACK) to be sent back to the MP. If the verification is successful, the RV will generate a signed TD packet and send it back to the MP. Caching of the TD should be forbidden for security concerns.

To verify the TI, the RV verifies the TI's signature using the MP's TI signing key corresponding to the tracing prefix, and also checks the timestamp against its own local clock with a reasonable grace period to detect replayed TIs that contain earlier timestamps. To make replay attack impossible, the RV needs to record the latest timestamp received from the MP and only accept TIs with strictly incremental timestamps. If the TI verification fails due to improper timestamp and the RV responses with a NACK, the MP may reinitiate the timestamp query.

Figure 3 demonstrates how a forwarder processes a TD and sets up trace. Upon receipt of a TD, a forwarder first treats it as a regular Data packet by 1) matching it against PIT entries and 2) forwarding it downstream if a match is found. Then, if a PIT match is found, the forwarder 3) searches for the "trace" tag in its name; if the tag is found, the 4) tracing prefix is extracted from the name by removing the tag and the verification information. The forwarder then 5) updates the FIB using the extracted tracing prefix and the incoming face of the corresponding TI for the received TD: if no FIB entry for the tracing prefix exists, a FIB entry is created for the tracing prefix with a next-hop list containing only the incoming face, or else the incoming face is merged into the existing entry's next-hop list; then a timer is set or adjusted for the newly created or refreshed next-hop to be removed in the future (§ 3.5).

KITE applies no restriction to how TIs are forwarded to eventually reach the RV, as traces are created with TDs; once TDs arrive at the MP, a trace exists between the RV and the MP. Hence, TIs may be forwarded with arbitrary forwarding strategies, and possibly along multiple paths.

The FIB entry created or updated during trace setup is referred to as "trace FIB entry", which contains a marker and co-exists with normal FIB entries created by the routing plane; alternatively, the forwarding information could be stored in a designated data structure, but it requires further changes to Interest forwarding.

The overall trace setup process forms a closed feedback loop, as the TD from the RV acts as feedback for the TI from the MP. The receipt of the TD implies the existence of two-way connectivity between the MP and the RV; otherwise, failing to receive the TD implies failures in the network or at the RV, and the MP is responsible for failure recovery. To enhance the tolerance to packet losses, one may allow the TDs be cached for a limited amount of time so that KITE may benefit from in-network caching for faster recovery from TD losses. However, TD caching should be enabled only in lossy environments because it compromises the security.

3.5 Trace Maintenance

There are two major challenges associated with the soft-state nature of the traces: *signaling overhead* and *stale trace*. Signaling overhead is generated when the MP periodically sends TIs to keep the traces alive when it is not moving; a stale trace is an old trace which leads to an old location of the MP, thus cannot guide Interests to the MP. If the traces are kept alive longer, the signaling overhead is reduced because the MP may send TIs less frequently; however, there is also a higher chance that a consumer's Interest is forwarded along a stale trace and eventually times out.

The *lifetime* of a trace is determined by the MP. We assume that MPs will not abuse the trace lifetime mechanism, otherwise, forwarders can arbitrarily limit the trace lifetime. The MP sets the lifetime value L and puts it in a TI as an Interest parameter [26]. After a next-hop H is created at time T, a forwarder sets a timer for H to expire and be removed at time $T + L$. The lifetime for H will be prolonged, if a TD is received before expiration and the corresponding TI (from the same face as H) carried a lifetime greater than the remaining lifetime.

To refresh the trace when the MP stays at the same location, the MP schedules the transmission of every next TI based on the lifetime carried in the previous one, i.e., the MP sets a *trace refreshing timer* to the lifetime of the last sent TI. Ideally, each MP should set the lifetime to the timeframe until its next relocation, so that only a single TI is sent until the MP relocates. In practice, the MP can leverage an estimation for the time of stay at its current location, e.g., the average period of previous stays.

The possibility that a trace becomes stale can be reduced by adaptive adjustment to the remaining lifetime. If the MP estimates that the relocation will occur in time L from now before the trace expires, it can send a new TI that carries the new lifetime L; and a forwarder will reduce the lifetime of the next-hop accordingly. Moreover, a to-be-stale trace can be actively purged by sending a TI that carries a lifetime of zero. The issue with stale traces is also addressed in 3.6.

The trace refreshing timer impacts tolerance to failures, because a longer timer means it generally takes longer to fix the broken trace due to topology/routing dynamics. Such failures are masked in the sense that, once the RV becomes reachable again, the MP can re-establish the trace to the RV by performing a new round of trace setup. To promptly recover from such failures, the TI refreshing timer needs to be set adequately short.

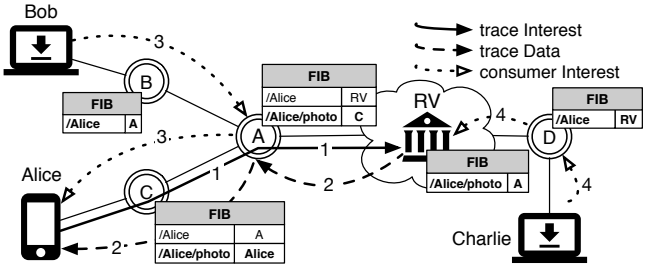


Figure 4: Consumer Interest forwarding and path shortcut

3.6 Consumer Interest Forwarding

Figure 4 shows the forwarding process of the consumer's Interests. The RV makes a routing announcement for "`/Alice`", which sets up a FIB entry at all forwarders for the routing prefix "`/Alice`". After the trace setup process, a trace FIB entry for the tracing prefix "`/Alice/photo`" is added to the FIB of forwarders A and C, and the RV. Bob and Charlie send Interests under the tracing prefix. On each forwarder, longest prefix match is used to find the FIB entry for Interests. Charlie's Interest will be first forwarded to the RV and then along traces to the MP, while Bob's Interest will be forwarded toward the RV, and before reaching the RV, it will be forwarded along the trace by forwarder A. In the case that there are multiple available next-hops in the matched FIB entry, the forwarding behavior is decided by specific forwarding strategies: forwarding Interests via all interfaces, only the last updated one, or one by one.

The forwarding behavior in the case of Bob's Interest reflects the *path shortcut* property of KITE that allows establishing a path to the MP not through the RV, which occurs if the MP and the consumer are on the same branch of the shortest-path tree sourced at the RV. Therefore, KITE remits one of the worst cases, where the path between two nearby nodes is stretched to the RV. In other words, if the MP and the consumer are in the same domain, path shortcut keeps the communication local even if the RV is out of the domain.

If a consumer Interest subject to path shortcut runs into a stale trace, PIT entries for this Interest will time out, which will be detected by forwarders on this stale trace. Note that forwarders may judge whether an Interest times out according to observed RTTs instead of the Interest lifetime. A forwarders may receive a NACK indicating that the Interest can not be satisfied through an outgoing interface. In both cases, forwarders may resort to a *forwarding strategy for the tracing prefix*. First, resend the Interest via other not-yet-tried outgoing interfaces in the matched FIB entry for the tracing prefix if available. Otherwise, resend the Interest through a not-yet-tried outgoing interface if two conditions are satisfied: *C1*) the interface is in a FIB entry for a shorter prefix, which is supposed to be the routing prefix, and *C2*) the interface is not the incoming interfaces nor previously tried. Otherwise, the Interest is discarded and a NACK may be sent back. The strategy is formally described in Algorithm 1.

A typical forwarding process with the proposed strategy is demonstrated in Figure 5: a consumer Interest 1) meets the stale trace at node B; node B then 2) forwards the Interest along the stale trace; after detecting the timeout, node B will 3) try to forward the Interest toward the RV; the Interest 4) meets a newer trace at node A and eventually reaches the MP. If the newer trace also becomes stale in the process, as long as all the forwarders adopt the forwarding strategy, the Interest can be eventually delivered to the MP. Note that except node B, no forwarder on the stale trace will resend the Interest, because the condition C2 is not satisfied.

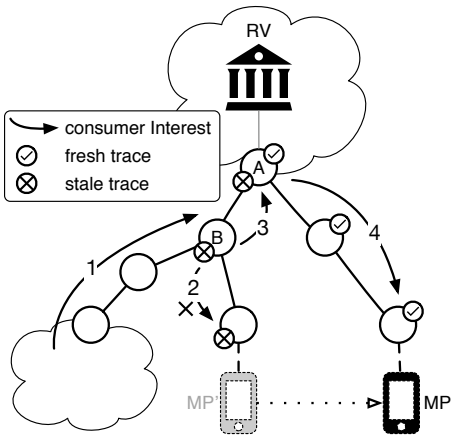


Figure 5: Mitigating the stale trace issue with forwarding strategy

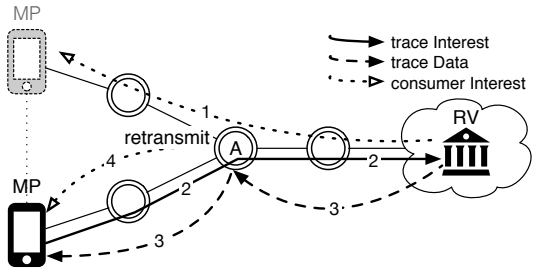


Figure 6: Retransmitting Interest if move-before-get

3.7 In-network Interest Retransmission

After the MP leaves a PoA, Interests for the MP will still be forwarded to this PoA before the traces are updated. For the consumer, these Interests are considered lost because they will never be satisfied, and need to be retransmitted. As suggested by Zhang et al. [34], the packet loss due to mobility can be recovered by utilizing *In-network Interest Retransmission (IIR)*, that allows forwarders to retransmit Interests if new forwarding information is available. As is shown in Figure 6, after forwarder A receives the TD, all the pending Interests under the tracing prefix are sent, or retransmitted, through the incoming face of the corresponding TI. IIR can mitigate Interest packet loss due to mobility, especially in the case of localized incremental movements, thus reducing handover delay (§ 6).

Note that a forwarder performs IIR on consumer Interests only if the new trace creates a new next-hop in the FIB entry; so IIR only happens at the junctional forwarder (forwarder A in Figure 6) of the old and new traces. Also, to avoid excessively large numbers of

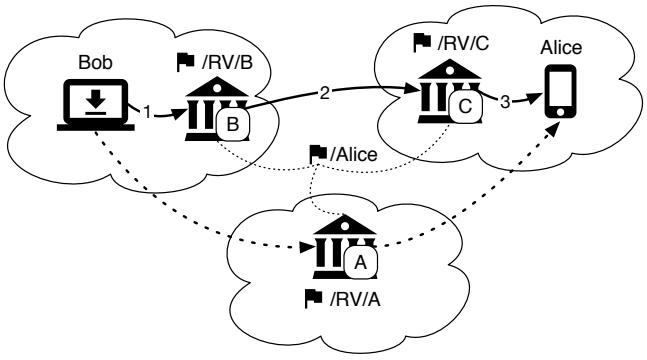


Figure 7: The multi-RV example

pending Interests being retransmitted simultaneously, IIRs will be spaced out in time with an upper bound.

3.8 Distributed Rendezvous

Distributed rendezvous with multiple cooperative RV servers makes it possible to scale KITE with the number of MPs, while other benefits include improving performance by mitigating triangular path problem (see simulation results in § 6.2.5), and enhancing the robustness by eliminating a single point of failure.

We propose multi-RV KITE, a preliminary design of distributed KITE. Multiple RV servers are deployed, and announce the same routing prefix; each server also announces its own unique *instance prefix*. Similar to IP anycast [19], Interests under the routing prefix, including TIs and Interests for the MP, are forwarded toward the nearest RV instance. The closest RV is called *attachment RV* for the MP, and *access RV* for the consumer.

In Figure 7, we show an example deployment setup of multi-RV KITE. Bob (consumer) retrieves data from Alice's mobile phone (MP). The RV instances A, B and C keeps the attachment information of Alice's phone up-to-date, and B (access RV) sets the forwarding hint [1] of Bob's Interests to "`/Alice/C`", the instance prefix of C (attachment RV). The Interests are then forwarded according to the forwarding hint to C. At C, the Interests will meet the trace and be further forwarded to reach Alice's phone.

The attachment information of the MPs' prefixes can be shared among the instances in real-time by running synchronization protocols proposed for distributed dataset synchronization in NDN, such as PSync [31], VectorSync [24] or RoundSync [6]; or with periodic updates backed by event-driven Interest relaying between old and new attachment RVs. Regarding attachment information inconsistencies among RV instances, consumer Interests may be relayed or buffered to reduce packet losses.

4 PROOF OF REACHABILITY

In this section, we prove that KITE guarantees that consumer Interests with the tracing prefix, or simply Interests, eventually reach the MP, as long as the MP receives TDs from the RV. First, we prove that there exists at least one forwarding path for the tracing prefix from any node to the MP. Then, we prove that Interests will reach the MP by using the forwarding strategy in § 3.6.

ALGORITHM 1: Forwarding Strategy For Consumer Interests

Input: An Interest I with the prefix p_t incoming from a neighbor c.
Output: A packet to be sent back to c.

```
1  if this node is the MP then return Data;
2  if not the first time receiving I then return NACK;
3  FIFO_Queue OutgoingQueue = [] ;
4  foreach ⟨p, n → h⟩ in the FIB in descending order of length (p) do
5      if p is the prefix of p_t and h ≠ c then
6          enqueue (OutgoingQueue, h) ;
7  while OutgoingQueue ≠ [] do
8      h ← dequeue (OutgoingQueue) ;
9      forward I through h ;
10     switch receive a packet do
11         case Data for I from h do return Data ;
12         case NACK for I from h do continue;
13         case I from node k ≠ h do  send NACK to k; goto Line 10 ;
14 return NACK ;
```

4.1 The Forwarding Path To The MP

Let the network be a graph containing a set of nodes N and a set of links L. An interface at a node is connected to a single neighboring node. Consider a general case that there are one node n_{rv} as the RV and one node n_{mp} as the MP, and any node $n \in N$ can be a consumer. Let p_r denote the routing prefix which is announced by n_{rv} in the routing plane. Let p_t denote the tracing prefix. Let a FIB entry e at a node n be a tuple $\langle p, n \rightarrow h \rangle$, where p is a name prefix, $(n, h) \in L$, and h is the next-hop for p. For the same prefix at the same node, let different FIB entries denote different next-hops. Given two nodes n, m and a prefix p, let $f(p, n \rightsquigarrow m)$ denote a forwarding path from n to m for p, which is defined below.

Definition 4.1. A forwarding path $f(p, n \rightsquigarrow m)$ of length $l > 0$ is a sequence of FIB entries $[e_1, e_2, \ldots, e_l]$, where $e_i = \langle p_i, n_i \rightarrow h_i \rangle$ for $i = 1, 2, \ldots, l$, satisfying the following conditions:

- *p-covered*: $\forall p_i$, p_i is the prefix of p (including $p_i = p$);
- *Sequential*: $n = n_1$, $m = h_l$, and $\forall i < l$, $h_i = n_{i+1}$;
- *Loop-free*: $\nexists n_i = n_j$ for $i \neq j$, and $\nexists n_i = m$.

We assume that the routing prefix announced by the RV is globally reachable.

Assumption. $\forall n \in N : \exists f(p_r, n \rightsquigarrow n_{rv})$.

Upon receipt of a TD message, the MP is reachable along the trace from the RV.

Lemma 4.2. *$\exists f(p_t, n_{rv} \rightsquigarrow n_{mp})$ if n_{mp} receives a TD message.*

Proof. Let $f(p_r, n_{mp} \rightsquigarrow n_{rv})$ of length l be one of forwarding paths along which the corresponding TI message reached n_{rv}. Given $\forall e_i = \langle p_i, n_i \rightarrow h_i \rangle \in f(p_r, n_{mp} \rightsquigarrow n_{rv})$, $i = 1, 2, \ldots, l$, KITE creates a FIB entry $g_{(l-i+1)} = \langle p_t, h_i \rightarrow n_i \rangle$. Then we have $f(p_t, n_{rv} \rightsquigarrow n_{mp}) = [g_1, g_2, \ldots, g_l]$. □

Theorem 4.3. *$\forall n \in N$, $\exists f(p_t, n \rightsquigarrow n_{mp})$ if n_{mp} receives a TD message.*

Proof. According to the assumption, $\exists f(p_r, n \rightsquigarrow n_{rv}) = [e_1, e_2, \ldots, e_l]$, and according to Lemma 4.2, $\exists f(p_t, n_{rv} \rightsquigarrow n_{mp}) = [g_1, g_2, \ldots, g_q]$.

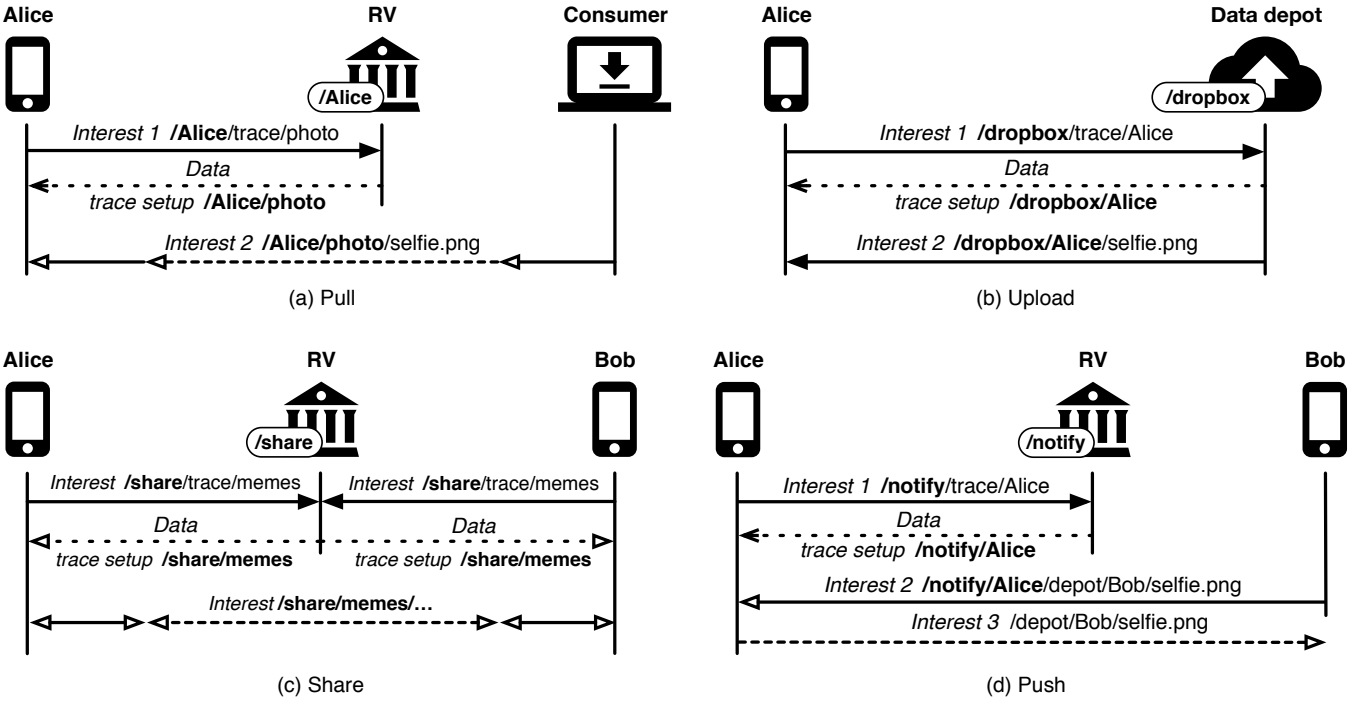


Figure 8: Message exchanges in mobile application protocols. Bold prefixes are in FIBs.

- Case1: If $\exists e_u = \langle p_r, n_u \rightarrow h_u \rangle$ such that $h_u = n_{mp}$, then obviously $\exists f(p_t, n \rightsquigarrow n_{mp}) = [e_1, \ldots, e_u]$.
- Case2: If $\exists e_u = \langle p_r, n_u \rightarrow h_u \rangle$, and $g_v = \langle p_t, n_v \rightarrow h_v \rangle$ such that $n_u = n_v$, then let s be the smallest u. Otherwise, let $s = l + 1$ and $v = 1$. We have $f(p_t, n \rightsquigarrow n_{mp}) = [e_1, \ldots, e_{(s-1)}, g_v, \ldots, g_q]$ which is p_t-covered, sequential and loop-free (details omitted).

□

4.2 Consumer Interests Forwarded To The MP

The forwarding strategy for consumer Interests presented in § 3.6 is formally described in Algorithm 1. The incoming consumer Interest will be sent through next-hop neighbors (Line 4-6) *strictly* one by one (Line 7-9). Assume no packet loss. Assume that a *virtual* NACK packet will return (Line 10), if an Interest packet is sent to a non-existing neighbor, such as an MP that has moved away. A forwarder will generate a NACK packet (Line 2, 13, 14), if the forwarder has seen the Interest before (Line 2), there is a loop (Line 13), or there is no more available next-hop (Line 14). Otherwise, a Data packet must be received (Line 1, 11).

Let a directed graph T consist of links derived from all FIB entries for the routing prefix and the trace prefix (Line 4-6). A FIB entry $\langle p, n \rightarrow h \rangle$ becomes a directed link (n, h) in T. Algorithm 1 is equivalent to a *recursive* depth-first search for the MP with the consumer as the source on T. When the Interest arrives at a node for the first time, the node is *visited* and can not be visited any more (Line 2, 13). Forwarding the Interest to a neighbor and waiting for a response is equivalent to a recursive function call (Line 9-10). Before every nodes on T is visited, the algorithm may terminate in visiting the MP (Line 1).

Therefore, by depth-first search, consumer Interests will reach the MP, as long as there exist paths from consumers to the MP on T, which can be guaranteed by Theorem 4.3 if the MP receives a TD message.

5 SCENARIO-AWARE PROTOCOL DESIGN

In this section, we demonstrate how to apply KITE to application protocol design in four typical mobile communication scenarios, as shown in Figure 8. Those scenarios differ in whether the MP initiates trace setup proactively or on demand, and whether the MP also acts as a mobile consumer. Therefore, the protocol design should be *scenario-aware*, which means that each scenario requires specific Interest/Data naming convention, as mentioned in § 3.2, as well as specific message exchange design. Note that KITE is still a network layer mechanism independent from applications.

5.1 Pull Scenario

In the Pull scenario, a consumer actively retrieves data from an MP. For instance, a consumer retrieves Alice's data under the tracing prefix "`/Alice/photo`" from Alice's mobile phone. The message exchanges are shown in Figure 8a. The RV is Alice's "home server", and announces Alice's routing prefix "`/Alice`". The MP proactively builds the trace with the RV, with Interest 1 being the first TI, and maintains the forwarding path by periodically sending TIs to the RV. Interests for Alice (such as Interest 2) will be forwarded toward the RV, and then follow traces to the MP.

5.2 Upload Scenario

In the Upload scenario, an MP uploads data to an immobile server. Typically, the MP may create the trace on demand, and then a consumer fetches data from the MP in the same way as the Pull scenario. In Figure 8b, we show how KITE supports uploading to a data depot which plays the role of RV. Alice wants to upload her selfie from her mobile phone to a data depot, e.g., a Dropbox uploading server, announcing its own prefix "`/dropbox`" (routing prefix). The tracing prefix identifies Alice under the data depot's namespace, e.g., "`/dropbox/Alice`". The TI may carry the exact name of the data, e.g., "`/dropbox/Alice/selfie.png`", to be uploaded as Interest parameters [26], which is not shown in Figure 8b. First, the MP sends Interest 1, which acts as a TI and an upload request. The depot obtains the name of the data to retrieve from Interest 1, and sends Interest 2 to retrieve the data.

5.3 Share Scenario

In the Share scenario, a group of MPs share data with each other. Each MP acts as both a producer and a consumer. We first consider the case of two MPs. In our example (Figure 8c), Alice and Bob have data to share with each other under the same prefix. They use the same RV announcing the routing prefix "`/share`" which is configured as a multicast prefix in the routing plane. The tracing prefix identifies a share group, e.g., "`/share/memes`". Each MP issues a TI to the RV; a TD is individually sent back for each valid TI, setting up traces for the tracing prefix to each MP. In the case of more than two MPs, the traces will form a forwarding tree rooted at the RV, which connects all MPs as leaves. By employing multicast forwarding for the "`/share/memes`" prefix, Interests matching the tracing prefix will be forwarded to all the MPs.

5.4 Push Scenario

In the Push scenario, an Interest sent to an MP is not a request for data, but a notification of some data ready to be fetched by the MP. The MP, actually as a consumer, may retrieve data according to this information, as if data is "pushed" to it. The notifications carry application-specific information in the name or as Interest parameters. In our example as shown in Figure 8d, Bob wants to push his selfie to Alice's phone. The RV announces the routing prefix "`/notify`", which identifies itself as Alice's notification service provider, and Alice receives Interests under the tracing prefix "`/notify/Alice`". Alice's phone periodically sends a TI, Interest 1, to proactively set up the traces. To notify Alice about a new piece of data to fetch, Bob issues a notification, Interest 2, whose name contains the name of the data. Upon receipt of Interest 2, Alice retrieves the data by sending Interest 3 with a name derived from Interest 2.

6 EVALUATION

The performance of KITE was evaluated in two ways: first, KITE was benchmarked against a conceptual mapping-based scheme through numerical experiments; second, a proof-of-concept implementation of KITE was evaluated through simulations.

Note that our purpose of evaluation is to investigate whether KITE can support producer mobility in various scenarios with an acceptable performance. The performance optimization is not considered as the design goal in this paper. We did not experimentally compare KITE with routing-based solutions, such as MAP-ME [3], because KITE is routing-transparent and the evaluation of performance related to routing dynamics and scalability is out of scope of this work.

6.1 Numerical Evaluation

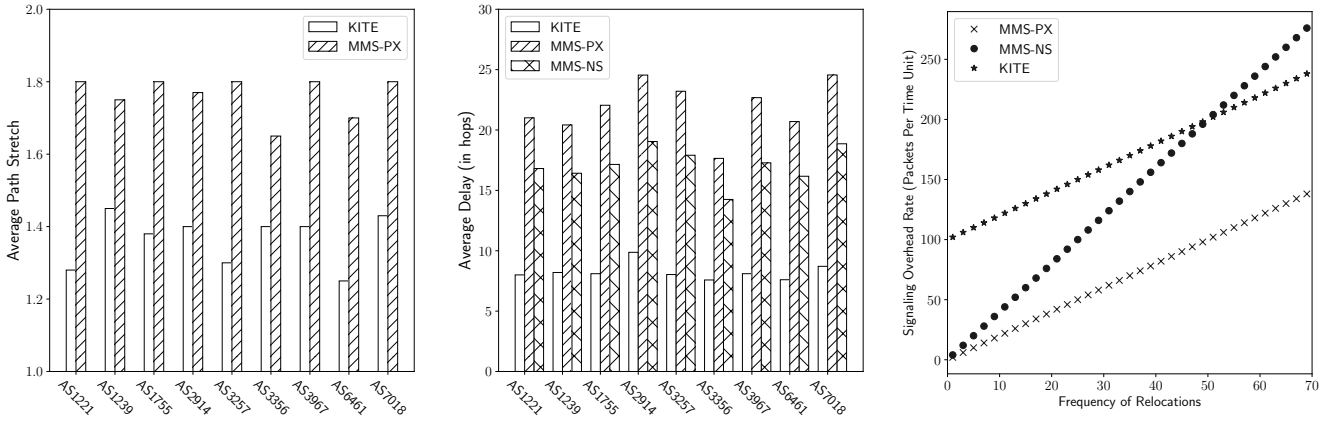
6.1.1 Methodology. Based on previous related work (§ 2), we abstract a conceptual scheme, Mapping-based Mobility Support (MMS), as the benchmark scheme. MMS uses explicit locators and name-to-locator mapping. In MMS, an MP learns and reports its current routable locator to an immobile mapping server by sending a locator update message. The mapping server can offer: 1) name resolution service, namely MMS-NS, which consumers leverage to obtain the MP's locator, or 2) proxy service, namely MMS-PX, that redirects the Interests from the consumer to the MP through tunneling. MMS also does not interact with the routing plane.

Our results are based on numerical measurements conducted on Rocketfuel topologies [25]. For each ISP's router-level topology, we run $0.1 \times N^2$ experiments, where N is the number of nodes. In each run, the access router of MP, the consumer, the RV and the mapping server are chosen independently and uniformly at random.

6.1.2 Path Stretch. Path stretch is the length ratio of the actual path to the shortest path between the MP and the consumer. We only compare the stretch of KITE with that of MMS-PX, as the stretch in MMS-NS is obviously 1. For the topologies of nine ISPs, the average length of shortest paths is 5.28-7.03 hops. As shown in Figure 9a, the average stretch of KITE is 1.26-1.46, about 20% less than that of MMS-PX (1.67-1.80). This result is expected, because KITE can take advantage of path shortcut (§ 3.6), while MMS-PX cannot. The results also show that path shortcut can remit the worst cases that the path between two nearby nodes is stretched to the RV. For example, for 86-98% of cases with stretch greater than 3 in MMS-PX, path shortcut can help to reduce the average stretch from 3.72-4.04 to about half (1.48-2.17).

6.1.3 Handover Delay. Handover delay is defined as the amount of time needed for the MP to receive pending Interests after relocation. The time of packet transmission is measured on the number of hops that the packet traverses. In KITE, pending Interests can be re-expressed by junctional forwarders with IIR (§ 3.7), so the handover delay depends on the distance in hops between the junctional forwarder and the MP, which may depend on the distance the MP moves. In our experiment, the MP moves to a random new location independent from the previous one, which is the worst case for KITE. For MMS-NS, in the best case that the consumer has a routable prefix, upon receipt of a locator-updating message from the MP, the mapping server can notify the consumer to resend expired Interests. Figure 9b shows that the average delay in KITE is in ~7.58-9.87 hops, which is ~38-43% of that in MMS-NS and 47-53% of that in MMS-PX.

6.1.4 Signaling Overhead. Signaling overhead is defined as the number of packets needed to signal the reachibility of an MP. Ideally, every time the MP relocates, two packets, a TI and a TD, are necessarily transmitted to set up a new trace in KITE. While staying



(a) Average path stretch **(b) Average handover delay** **(c) Signaling overhead rate**

Figure 9: KITE benchmarking results

at the same location, the MP has to keep sending TIs and receive TDs to refresh the soft-state trace. MMS needs at least one Interest/Data exchange (2 packets) for the MP to update its locator at the mapping server (MMS-PX), and one more Interest/Data exchange (2 more packets) to notify the consumer about the new locator (MMS-NS).

Figure 9c shows the rate of signaling packets as a function of the frequency of relocations, where the trace lifetime is constantly 1/50 of the time unit. When the MP is relatively stable, i.e., the trace lifetime is much shorter than the duration of stay at the same location, the overhead of KITE is higher than that of MMS. If the MP moves fast, no additional packets are needed to prolong the trace. In such cases, the KITE overhead is slightly better than the MMS-NS overhead. Note that these results refer to the worst case of overhead for KITE, since the MP does not reset the timer for refreshing the traces after each relocation.

In summary, compared with mapping-based solutions, KITE provides shorter path stretch due to path shortcut, better handover delay due to the IIR mechanism, and acceptable signaling overhead.

6.2 Proof-of-Concept Implementation

We implemented KITE on ndnSIM [17] as a proof-of-concept, including a version with multiple RVs that cooperate with each other, i.e., multi-RV KITE (§ 3.8). We evaluated the prototype in three application scenarios, Upload, Pull and Share, with a single RV, and the Pull scenario with multiple RVs. The source code is available at https://github.com/KITE-2018 [33].

6.2.1 Simulation Setup. The infrastructural network topology for simulation consists of 11 nodes with distance ~100m between each two, as shown in Figure 10. Each node functions as both Access Point (AP) and forwarder. The trace lifetime in TIs is set to a fixed value of 2 seconds, which is not adjusted adaptively. Upon relocating at a new AP, the MP sends a TI message immediately and periodically per 2 seconds afterwards. A TD message is cached for 100ms, and the MP will retransmit a TI message for up to 3 times by setting a transmission timer according to observed RTTs. In each run of simulation, the MP follows a random walk mobility pattern

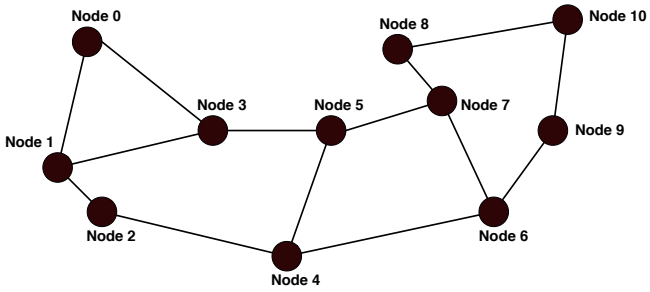
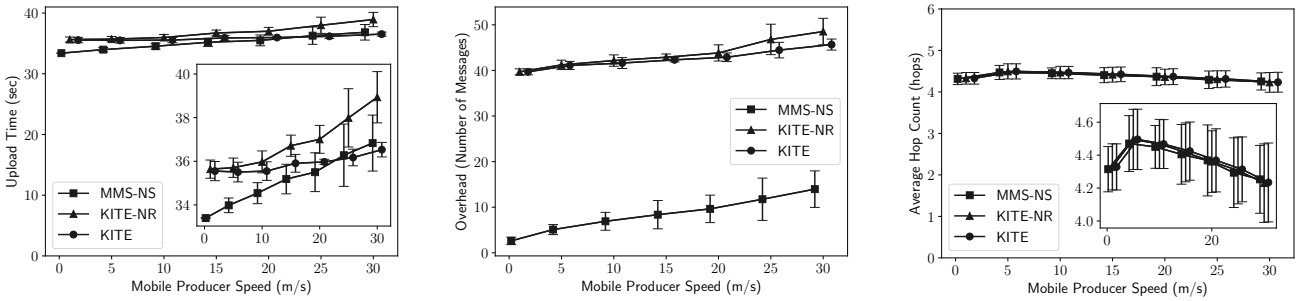


Figure 10: Infrastructural network simulation topology

by randomly changing the direction after moving 100m at a fixed speed within the area covered by the APs.

For the Pull and Upload scenarios with a single RV, we compare the performance of MMS-NS, KITE, and KITE without the IIR mechanism (§ 3.7) (KITE-NR). Specifically, in the Pull scenario (§ 6.2.2), the consumer continuously retrieves data from an MP for 100 seconds. The RV or the mapping server is connected to node 5. For each consumer placement throughout all nodes, we perform 5 runs for each different MP speeds. In the Upload scenario (§ 6.2.3), the MP uploads a file of size 1MB to a data depot, which plays both the roles of the RV/mapping server and the consumer. For each data depot placement throughout all nodes, we perform 5 simulation runs for each different MP speed. In the Share scenario (§ 6.2.4), we apply KITE to ChronoSync [37], an NDN protocol for decentralized data synchronization. For the Pull scenario with multiple RVs (§ 6.2.5), we compare the multi-RV KITE to MMS-NS. To simulate three RVs at node 1, 5, and 6 synchronizing on the MP's attachment information, we add a delay of 100ms for updating the MP's attachment information on all RVs.

6.2.2 Pull. Figure 12 presents the average results for the Pull scenario. Figure 12a shows that KITE performs better than both KITE-NR and MMS-NS, the later two have similar packet loss rates. The IIR mechanism enables KITE to recover packet losses during the



(a) Upload time vs. speed **(b) Signaling overhead vs. speed** **(c) Average hop count vs. speed**

Figure 11: Upload scenario - simulation results

MP's handover. Figure 12b shows that the MMS-NS signaling overhead increases as the speed increases, as a consequence of the increasing number of relocations. Because the consumer in this scenario does not have a routable prefix, the mapping server cannot directly notify the consumer when the MP updates its locator. Therefore, the consumer has to actively query the server when Interest timeouts occur under an assumption that losses are due to the MP's relocation. The KITE's overhead also increases as the speed increases, since the MP sends a new TI after each relocation. For high speeds though, the KITE overhead is lower than that of MMS-NS. Finally, Figure 12c shows that paths in KITE are ~13% longer than the shortest which is the case of MMS-NS.

6.2.3 Upload. Figure 11 presents the average results for the Upload scenario. Figure 11a shows that MMS-NS achieves shorter upload time than KITE-NP, while KITE slightly outperforms MMS-NS at higher speeds. Figure 11b shows that KITE and KITE-NP require more TI/TD messages than MMS-NS location updates, since the depot in MMS-NS directly offers the mapping server functionality (a single Interest/Data exchange is required per relocation). Finally, Figure 11c shows that KITE, KITE-NP and MMS-NS result in roughly equal path lengths.

6.2.4 Share. Figure 13 presents the average results for the Share scenario. KITE improves the scalability of ChronoSync by *multicasting* Interests among the mobile group members instead of *flooding* Interests over the whole network. We measured the total number of packets transmitted by forwarders and MPs. Compared to flooding, KITE reduces the total number of packets 17-31%. As the number of MPs increases, the reduction ratio increases as well.

6.2.5 Pull with Multi-RV. In Figure 14 presents the average results for the Pull scenario with multi-RV KITE and MMS-NS. Compared to MMS-NS, which represents the optimal path, the average path produced by multi-RV KITE is ~6.7% longer. The results show that the multi-RV solution can effectively mitigate the triangular path problem.

7 DISCUSSION

7.1 Architectural Implications

To the native NDN architecture, the only new ingredient introduced by KITE is the tracing mechanism as add-ons, rather than modifications, to the forwarding plane. Note that the tracing mechanism does not involve the process of packet forwarding, but manipulates FIBs according to local states. To minimize potential interference with other forwarding mechanisms, KITE is clearly modularized by tagging TI/TD signaling messages and restricting its workspace in FIBs to sub-prefixes under routing-specified prefixes. Nonetheless, there may be conflict if a FIB entry created by KITE is also operated by others, which should be avoided.

KITE further exploits the stateful forwarding plane. Upon receipt of TDs, forwarders transform the states of satisfied TIs in PITs into the states of traces in FIBs, so that mobility is handled by local decisions based on local states. Thus the forwarding plane keeps track of unstable locations of MPs, while the routing plane only needs to look after the reachability of immobile RVs. KITE also opens a new door to Interest multicast forwarding. For example, in the Share scenario, MPs directly build up the multicast tree with the RV in network layer.

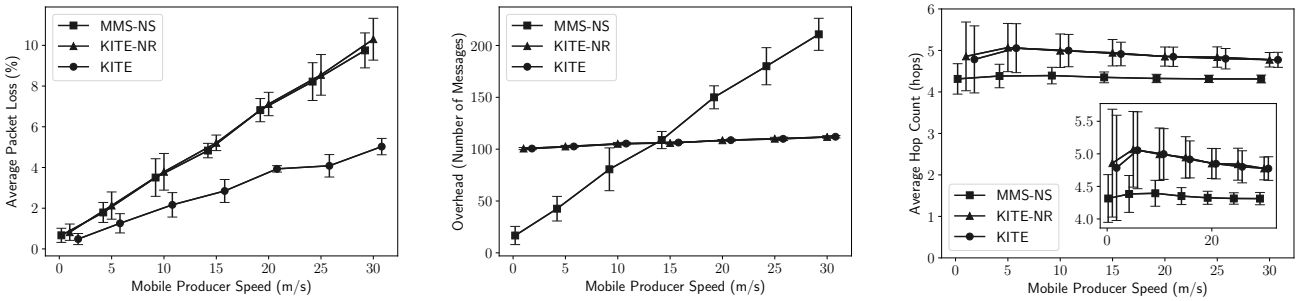
KITE's locator-freeness makes it feasible that a mobile or even immobile producer does not need any routing prefix bound with its location while providing data to consumers. In other words, the name of data is independent from the location of producer. In addition to the locator-freeness of consumer, a device can become completely locator-free in NDN. Furthermore, if tremendous amount of personal hand-hold/wearable devices and Internet-of-Things sensors provide data under the prefixes of data depots, such as "`/Facebook`", "`/Youtube`", or "`/Dropbox`", in the Upload scenario, they will not even need device-specific prefixes at all.

Except the benefits, the costs brought by KITE to all forwarders and the forwarding plane include potential security risks, additional states, processing overheads and complexity. A straightforward yet tough question one may ask is "*Is it worth the trouble?*". We believe the answer largely depends on the requirements of mobile applications in the future.

7.2 Security Considerations

The security of KITE is built upon three assumptions: 1) the whole routing plane and all forwarders are trusted; 2) RVs can not be compromised; 3) RVs and MPs know each other's public keys, no private key is leaked, and cryptography primitives, such as digital signatures, can not be broken.

The trace setup process is protected by authenticated TI/TD exchanges and the secure routing plane. An attacker may issue a fake



(a) Packet loss vs. speed **(b) Signaling overhead vs. speed** **(c) Average hop count vs. speed**

Figure 12: Pull scenario - simulation results

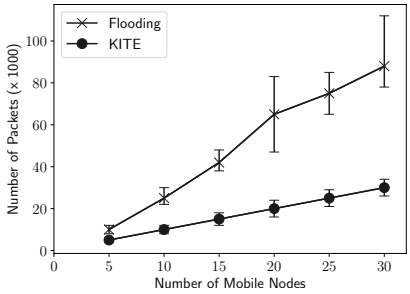


Figure 13: Share scenario - traffic volume vs. number of MPs

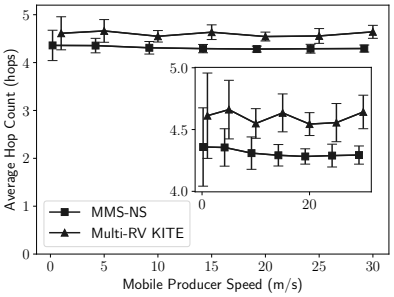


Figure 14: Multi-RV Pull scenario - hop count vs. speed

TI, but the RV will detect the fake one and do not reply a TD. Also, no fake TD can be pushed out, because the security of routing plane guarantees that the routing prefix can not be hijacked so that except the RV, no attacker will receive a TI with the routing prefix. By flooding the RV with massive amount of different fake TIs, attackers consume the RV's computing and bandwidth resources to carry out Denial-of-Service (DoS) attacks. In this case, the RV can detect fake TIs and utilize NDN's hop-by-hop pushback with NACK messages towards attackers, and trigger DoS mitigation mechanisms (such as Producer-Assisted Pushback, PAP [35]). The bottom line of security is that forged TI/TD messages can be detected.

In the current design, digital signatures are employed as authenticity proof. Compagno et al. [5] proposed a hash-chain-based "light-weight prefix attestation protocol" that can be applied to various producer mobility solutions ([3, 9, 13, 22, 34]), including the previous KITE design, that use special signaling Interest messages for reachability information update; the protocol enables forwarders to verify the validity and freshness of the signaling Interests without significant impact on regular packet processing. However, the proposed protocol also requires per prefix "security context" be propagated network-wide, the feasibility of which remains to be evaluated. Although we rely on authenticated Interest-Data exchanges to set up trace, the RV still needs to verify the authenticity of TIs; thus the hash-chain-based method can be adopted to reduce the computational costs and enhance the resilience to DDoS attacks, which belongs to our future work.

8 CONCLUSION

In this paper, we have presented KITE, a tracing-based producer mobility support solution for NDN. By establishing soft-state forwarding paths through authenticated Interest-Data exchange between mobile producers and routable rendezvous, KITE achieves locator-freeness, data-retrieval transparency, routing transparency and abuse-proof. We have applied KITE to the design of application protocols for various mobile communication scenarios. The performance of KITE has been evaluated by numerical and packet-level simulations. We believe that KITE will become an indispensable NDN building block that will further enhance the power of the stateful forwarding plane.

Our future work includes: 1) further improving the distributed rendezvous design for the scalability of KITE; 2) optimizing soft-state trace maintenance to reduce signaling overhead and mitigate the impact of stale traces; 3) refining the security measures for stronger security guarantee and better scalability; and 4) evaluating the feasibility for real-world deployment by quantifying the architecture implications.

9 ACKNOWLEDGMENTS

This work is partially supported by the National Key Research and Development Program of China (2016YFB0801303), National Science Foundation under awards CNS-1629922 and CNS-1719403, and State Scholarship Fund from China Scholarship Council (No. 201706120303).

REFERENCES

[1] Alexander Afanasyev, Cheng Yi, Lan Wang, Beichuan Zhang, and Lixia Zhang. 2015. SNAMP: Secure namespace mapping to scale NDN forwarding. In *Computer*

Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on. IEEE, 281–286.

[2] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machado, Wenfei Wu, Aditya Akella, David G Andersen, John W Byers, et al. 2011. XIA: An architecture for an evolvable and trustworthy Internet. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. ACM, 2.

[3] Jordan Augé, Giovanna Carofiglio, Giulio Grassi, Luca Muscariello, Giovanni Pau, and Xuan Zeng. 2018. MAP-Me: Managing Anchor-less Producer Mobility in Content-Centric Networks. *IEEE Transactions on Network and Service Management* (2018).

[4] Aytac Azgin, Ravishankar Ravindran, and Guoqiang Wang. 2014. A scalable mobility-centric architecture for named data networking. *arXiv preprint arXiv:1406.7049* (2014).

[5] Alberto Compagno, Xuan Zeng, Luca Muscariello, Giovanna Carofiglio, and Jordan Augé. 2017. Secure producer mobility in information-centric network. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 163–169.

[6] Pedro de-las Heras-Quirós, Eva M Castro, Wentao Shang, Yingdi Yu, Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. *The design of RoundSync protocol.* Technical Report. Technical Report NDN-0048, NDN.

[7] Deborah Estrin, Dino Farinacci, Ahmed Helmy, David Thaler, Stephen Deering, Mark Handley, Van Jacobson, Ching-Gung Liu, Puneet Sharma, and Liming Wei. 1998. *Protocol independent multicast-sparse mode (PIM-SM): Protocol specification.* Technical Report.

[8] A. Grilo, P. Estrela, and M. Nunes. 2001. Terminal independent mobility for IP (TIMIP). *IEEE Commun. Magazine* 39, 12 (2001), 34–41. https://doi.org/10.1109/35.968810

[9] D Han, M Lee, K Cho, TT Kwon, and Y Choi. 2012. Pmc: Publisher mobility support for mobile broadcasting in content centric networks. *ASIA Future Internet* (2012).

[10] Frederik Hermans, Edith Ngai, and Per Gunningberg. 2012. Global Source Mobility in the Content-centric Networking Architecture. In *NoM '12*.

[11] V. Jacobson et al. 2012. Custodian-based Information Sharing. *IEEE Communications Magazine* 50, 7 (2012), 38–43.

[12] Petri Jokela, András Zahemszky, Christian Esteve Rothenberg, Somaya Arianfar, and Pekka Nikander. 2009. LIPSIN: Line Speed Publish/Subscribe Internetworking. In *SIGCOMM '09*.

[13] Do-hyung Kim, Jong-hwan Kim, Yu-sung Kim, Hyun-soo Yoon, and Ikjun Yeom. 2012. Mobility Support in Content Centric Networks. In *ICN '12*.

[14] Deguang Le, Xiaoming Fu, and Dieter Hogrefe. 2006. A review of mobility support paradigms for the internet. *IEEE Commun. Surveys Tutorials* 8, 1 (2006), 38–51. https://doi.org/10.1109/COMST.2006.323441

[15] Jihoon Lee, Sungrae Cho, and Daeyoub Kim. 2012. Device mobility management in content-centric networking. *IEEE Commun. Magazine* 50, 12 (2012), 28–34. https://doi.org/10.1109/MCOM.2012.6384448

[16] Dawei Li and Mooi Choo Cuah. 2013. SCOM: A scalable content centric network architecture with mobility support. In *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*. IEEE, 25–32.

[17] Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. On the evolution of ndnSIM: An open-source simulator for NDN experimentation. *ACM SIGCOMM Computer Communication Review* 47, 3 (2017), 19–33.

[18] Jayanth Mysore and Vaduvur Bharghavan. 1997. A New Multicasting-based Architecture for Internet Host Mobility. In *MobiCom '97*.

[19] Craig Partridge, Trevor Mendez, and Walter Milliken. 1993. *Host Anycasting Service.* RFC 1546. RFC Editor. https://www.rfc-editor.org/rfc/rfc1546.txt

[20] C. Perkins. 2010. IP Mobility Support for IPv4, Revised. *RFC 5944* (2010).

[21] R. Ramjee, K. Varadhan, L. Salgarelli, S.R. Thuel, Shie-Yuan Wang, and T. La Porta. 2002. HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks. *IEEE/ACM Trans. on Networking* 10, 3 (2002), 396–410. https://doi.org/10.1109/TNET.2002.1012370

[22] Ravishankar Ravindran, Samantha Lo, Xinwen Zhang, and Guoqiang Wang. 2012. Supporting seamless mobility in named data networking. In *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 5854–5869.

[23] Dipankar Raychaudhuri, Kiran Nagaraja, and Arun Venkataramani. 2012. Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet. *ACM SIGMOBILE Mobile Computing and Communications Review* 16, 3 (2012), 2–13.

[24] Wentao Shang, Alexander Afanasyev, and Lixia Zhang. 2017. VectorSync: distributed dataset synchronization over named data networking. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 192–193.

[25] Neil Spring, Ratul Mahajan, and David Wetherall. 2002. Measuring ISP Topologies with Rocketfuel. In *SIGCOMM '02*.

[26] NDN Team. 2018. NDN Interest Packet Format - Parameters. http://named-data.net/doc/NDN-packet-spec/current/interest.html#parameters

[27] Gareth Tyson, Nishanth Sastry, Ruben Cuevas, Ivica Rimac, and Andreas Mauthe. 2013. A survey of mobility in information-centric networks. *Commun. ACM* 56, 12 (2013), 90–98.

[28] András G. Valkó. 1999. Cellular IP: A New Approach to Internet Host Mobility. *SIGCOMM Comput. Commun. Rev.* 29, 1 (1999), 50–65. https://doi.org/10.1145/505754.505758

[29] Liang Wang, Otto Waltari, and Jussi Kangasharju. 2013. Mobiccn: Mobility support with greedy routing in content-centric networks. In *Global Communications Conference (GLOBECOM), 2013 IEEE*. IEEE, 2069–2075.

[30] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, Patrick Crowley, Christos Papadopoulos, Lan Wang, Beichuan Zhang, et al. 2014. Named data networking. *ACM SIGCOMM Computer Communication Review* 44, 3 (2014), 66–73.

[31] Minsheng Zhang, Vince Lehman, and Lan Wang. 2017. Scalable name-based data synchronization for named data networking. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 1–9.

[32] Yu Zhang, Alexander Afanasyev, Jeff Burke, and Lixia Zhang. 2016. A Survey of Mobility Support in Named Data Networking. In *IEEE INFOCOM NoM '16*.

[33] Yu Zhang and Zhongda Xia. 2018. KITE Simulation with ndnSIM. https://github.com/KITE-2018

[34] Yu Zhang, Hongli Zhang, and Lixia Zhang. 2014. Kite: A Mobility Support Scheme for NDN. In *ACM ICN'14*. 179–180.

[35] Zhiyi Zhang, Vishrant Vasavada, Jonathan Lin, Reddy Siva Kesava K, Peter Reiher, and Lixia Zhang. 2018. *Producer-Assisted Pushback.* Technical Report. Technical Report NDN-0065, NDN.

[36] Zhiyi Zhang, Haitao Zhang, Eric Newberry, Spyridon Mastorakis, Yanbiao Li, Alexander Afanasyev, and Lixia Zhang. 2018. *Security support in Named Data Networking.* Technical Report. NDN Project, Technical Report. NDN-0057,(Mar. 2018).

[37] Zhenkai Zhu and Alexander Afanasyev. 2013. Let's chronosync: Decentralized dataset state synchronization in named data networking. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on*. IEEE, 1–10.

[38] Z. Zhu, R. Wakikawa, and L. Zhang. 2011. A Survey of Mobility Support in the Internet. *RFC 6301* (2011).