KITE: Producer Mobility Support in Named Data Networking

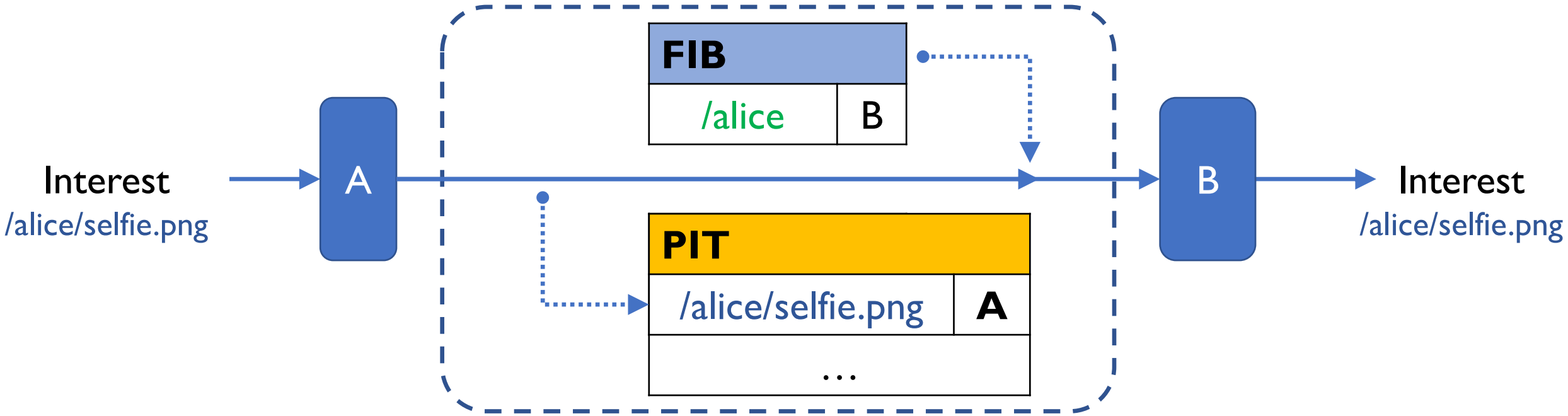
Yu Zhang[1], **Zhongda Xia**[1], Spyridon Mastorakis[2], Lixia Zhang[2]

[1] Harbin Institute of Technology [2] UCLA

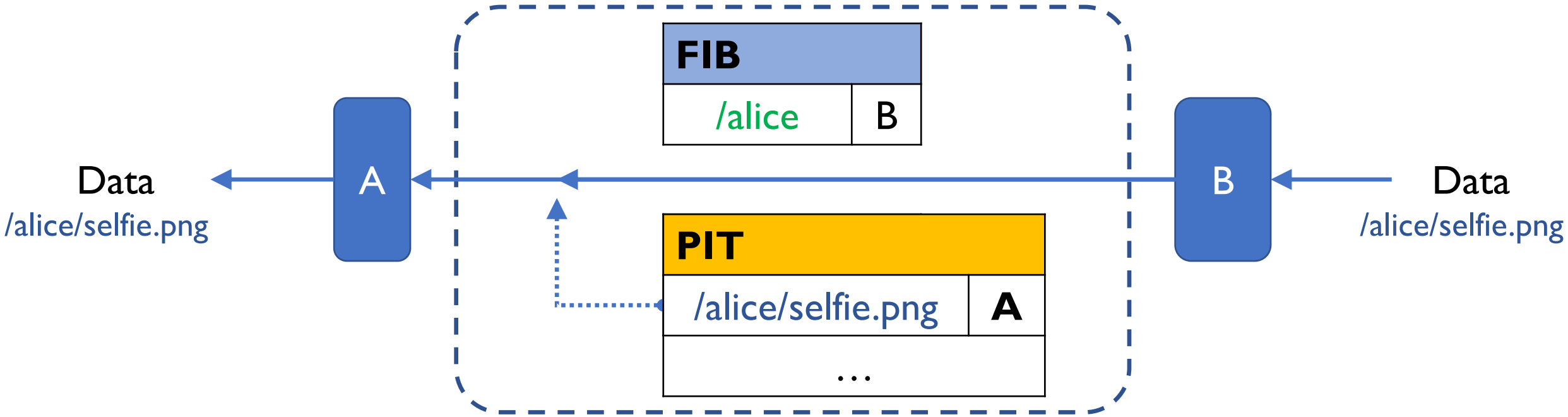
NDN Mobility Support

- **Consumer mobility** is natively supported ✓
  - pull-based communication model + stateful forwarding plane
- **Producer mobility** is still an open issue ✗
  - existing solutions are not "simple" enough
- **Motivation:** can we further exploit native NDN features to support producer mobility in a relatively simple way?

NDN Stateful Forwarding Plane

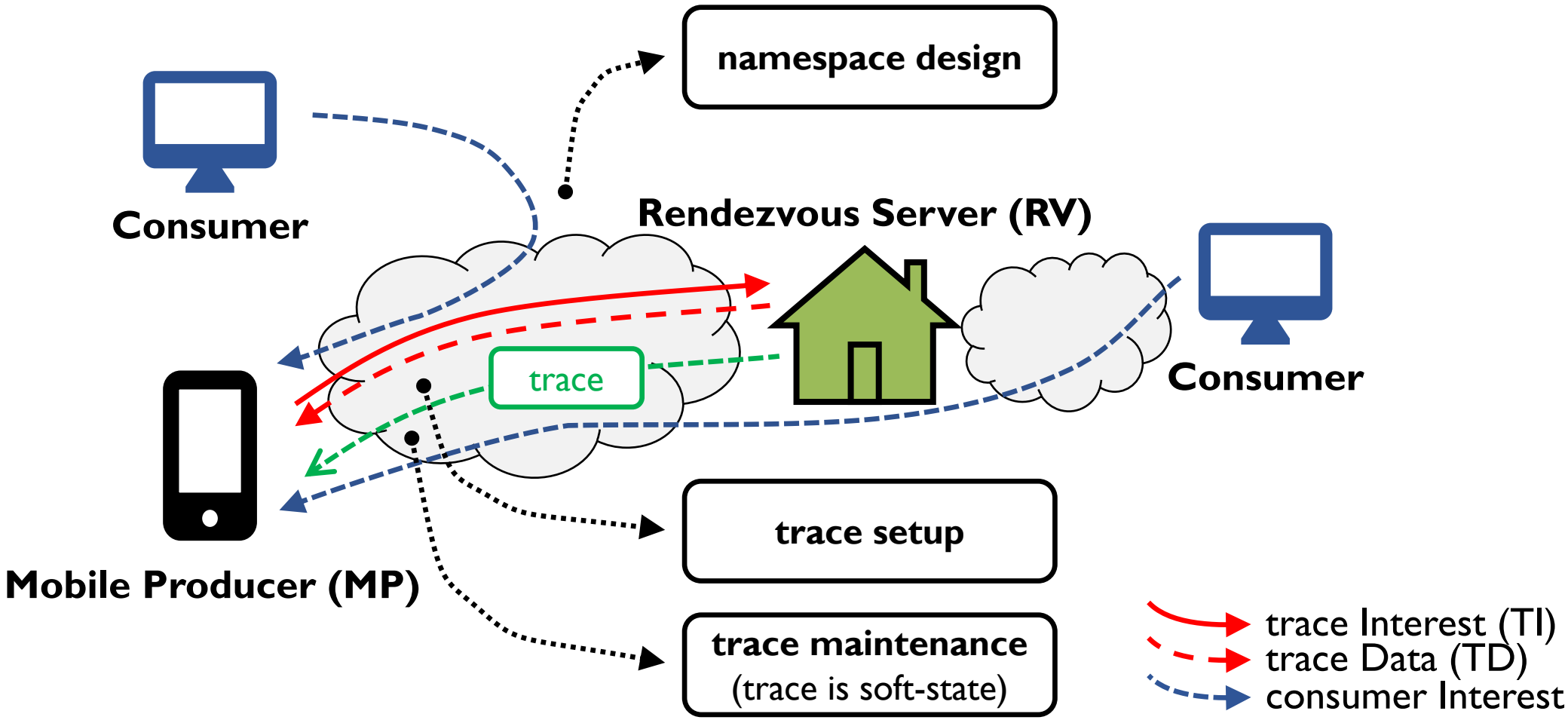


NDN Stateful Forwarding Plane

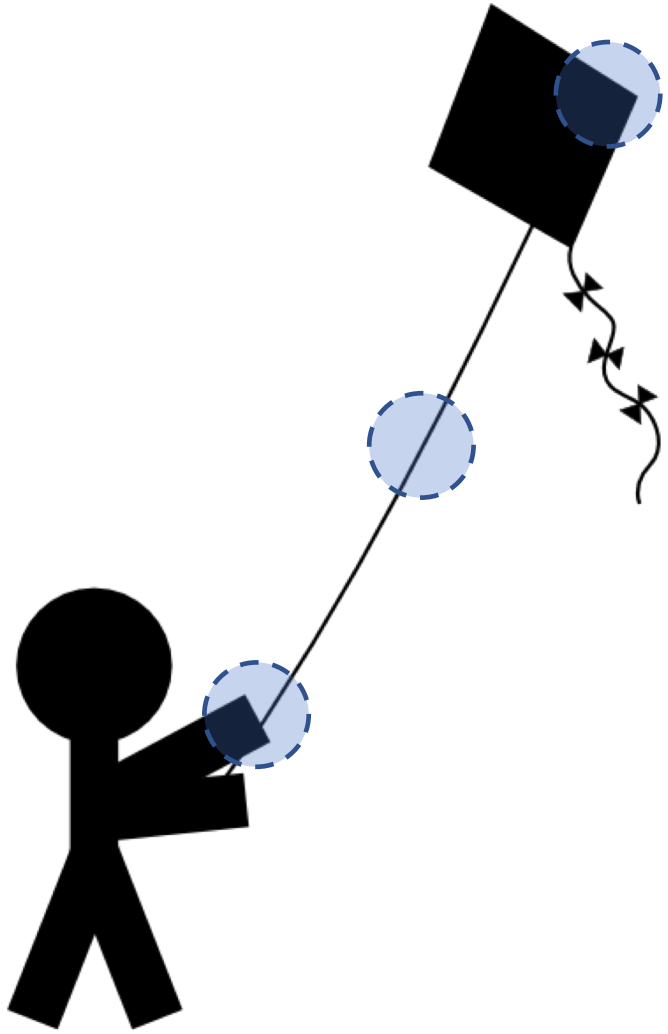


- Data is forwarded back along a hop-by-hop "breadcrumb trail"
  - KITE saves the "breadcrumb trail" for Interest forwarding

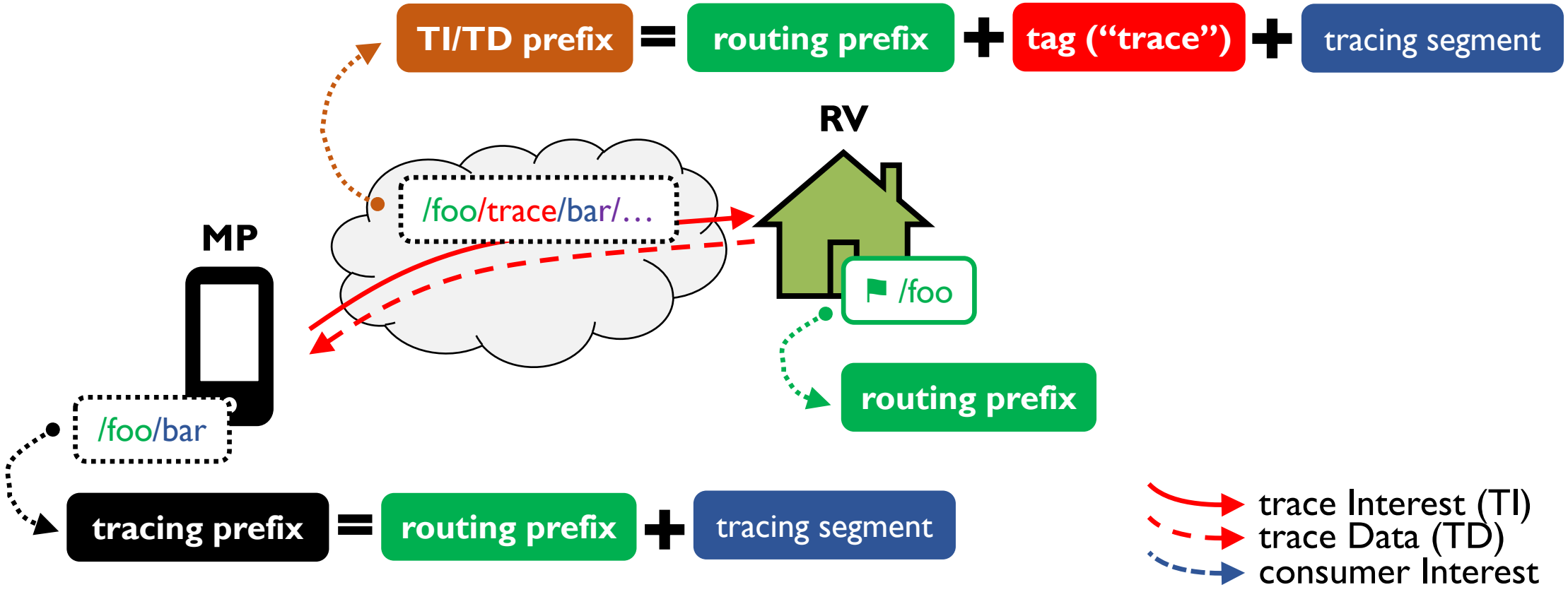
Overview


namespace design
Consumer
Rendezvous Server (RV)
Consumer
trace
Mobile Producer (MP)
trace setup
trace maintenance
(trace is soft-state)
trace Interest (TI)
trace Data (TD)
consumer Interest

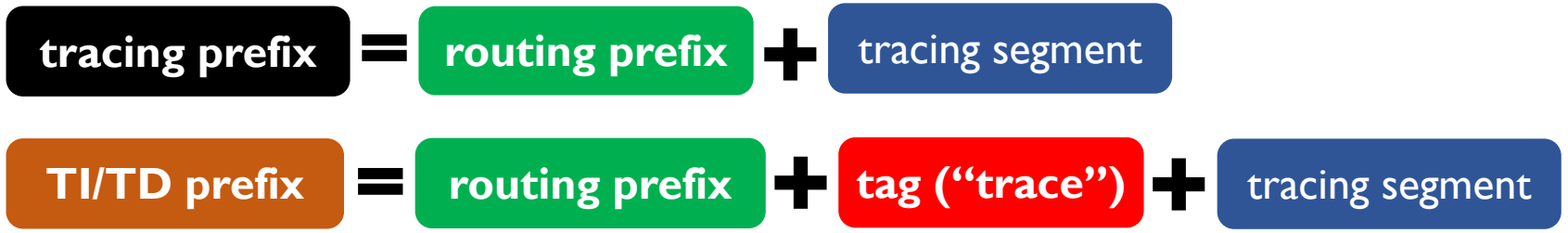
As simple as kite flying



Namespace Design

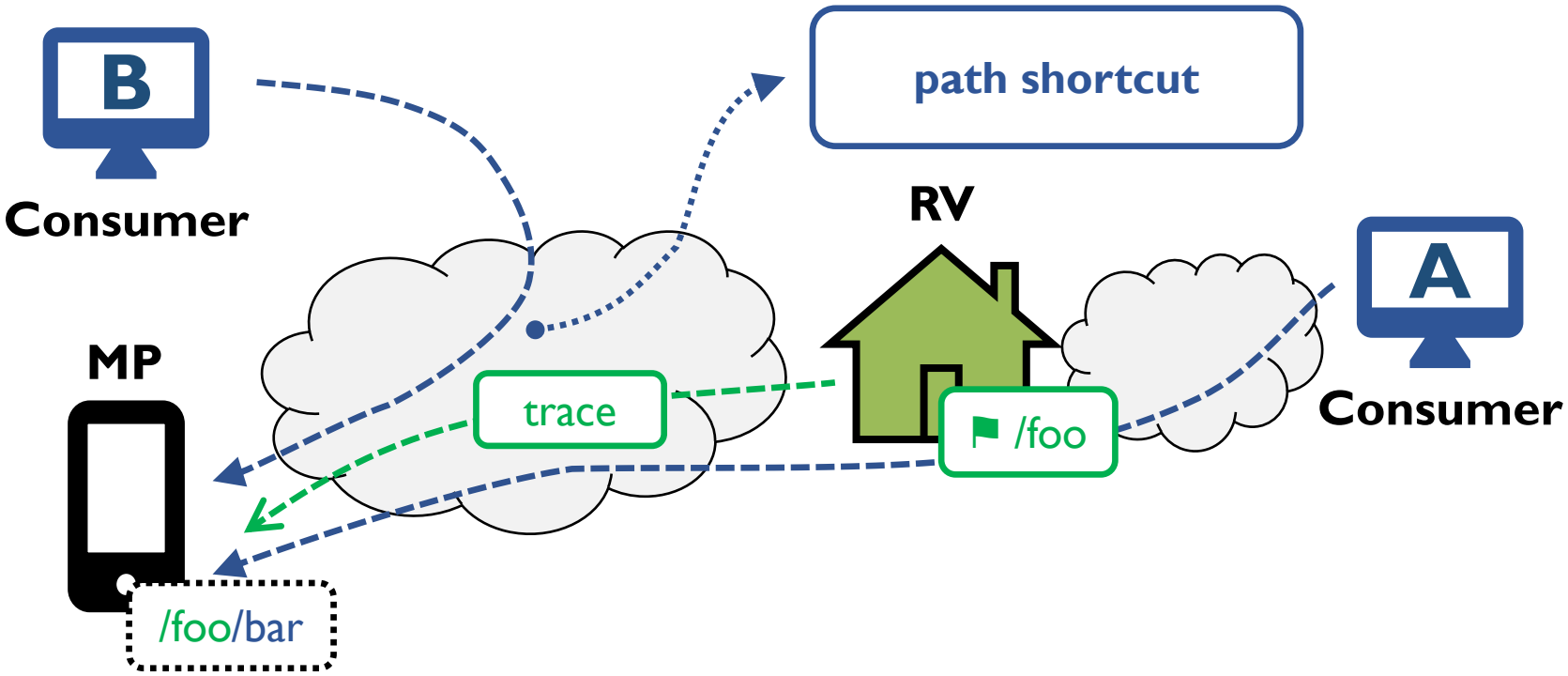


Namespace Design: Explained



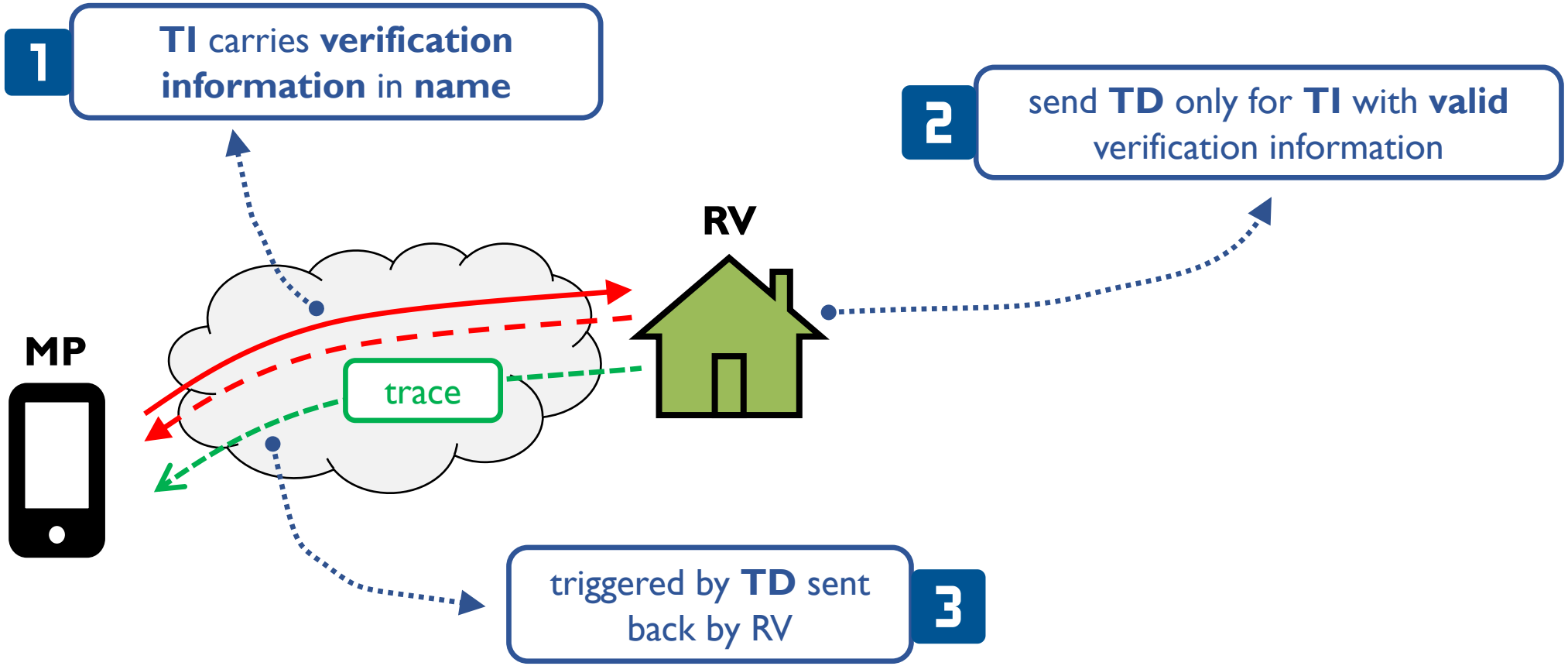
- **Tracing prefix** is longer than **routing prefix**, so trace setup is separated from the routing plane
- **Tracing prefix** and **TI/TD prefix** share a common prefix: **routing prefix**, so consumer Interest and trace can meet at the RV
- **Tracing prefix** can be derived from **TI/TD prefix** by removing **tag**
- **Tag** is a reserved keyword ("trace") that separates trace setup from data retrieval

Consumer Interest Forwarding



consumer Interest

Trace Setup


1
TI carries verification information in name
2
send TD only for TI with valid verification information
RV
MP
trace
triggered by TD sent back by RV
3

Trace Setup: Prevent Prefix Hijacking

- We make the following assumptions:
  - **(1)** signature cannot be spoofed
  - **(2)** RV is not compromised
  - **(3)** the infrastructure is trusted
- Prefix hijacking is prevented under the assumptions above
  - attacker cannot push TD out: not possible with NDN
  - attacker cannot generate valid TI: **(1)**
  - attacker cannot pull TD back by sending fake TI: **(2)**
  - attacker cannot receive TI and collude with a partner: **(3)**

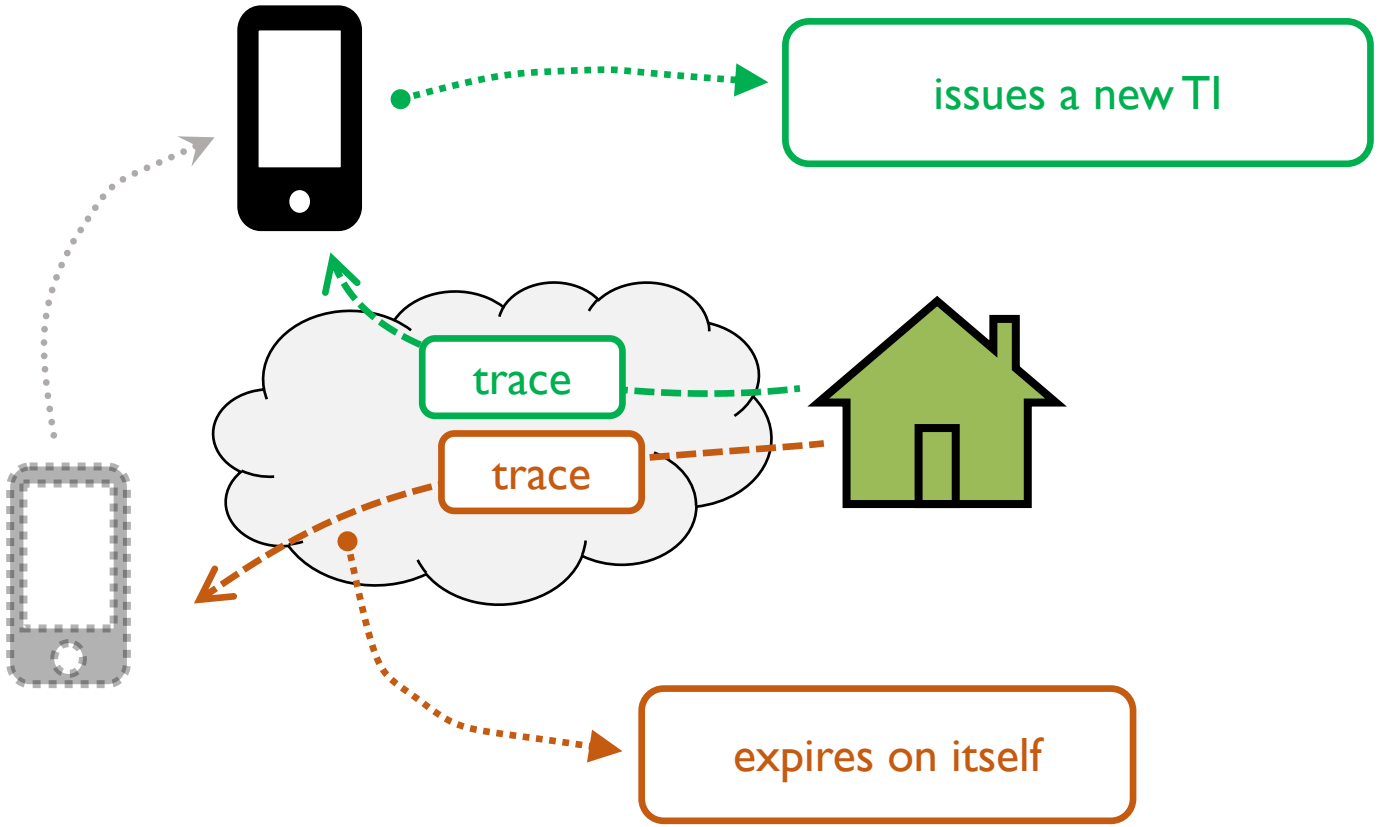
Trace Setup: Summary

- Forwarder forwards TI and TD as regular packets
- Forwarder searches for the "trace" tag to identify a TD
- Prefix hijacking is prevented by doing verification at the RV
- The trace setup process forms a closed feedback loop
  - receipt of TD implies two-way connectivity
  - MP can recover from packet loss during trace setup by retransmitting TI

Trace Maintenance

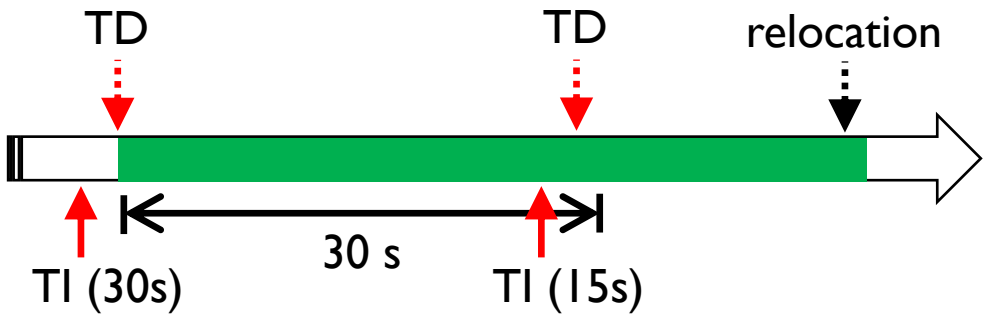
- Why soft-state?
  - Simple, no inter-forwarder protocol
- How?
  - MP decides the lifetime, and put in TI as “Parameter”
  - Forwarder obeys MP’s instruction and manages the lifecycle of trace accordingly

Soft-state Trace and Relocation



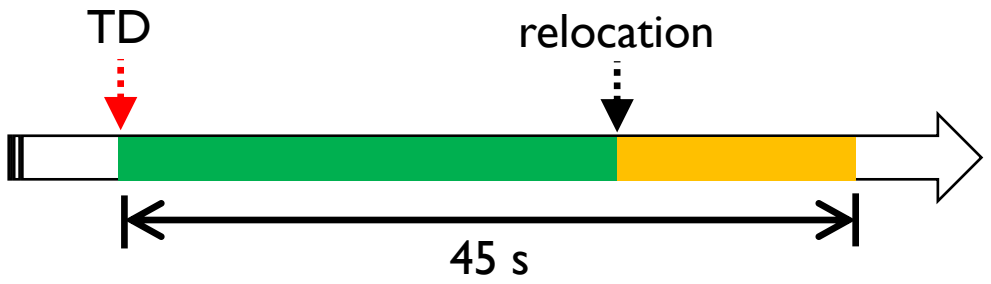
Issues with Soft-state: Signaling Overhead

- Extra signaling overhead is incurred if trace expires **before** relocation
  - more than one TI needs to be sent per relocation
  - MP may adjust lifetime setting according to estimated time of stay
  - lifetime should be set short enough to recover from failures promptly

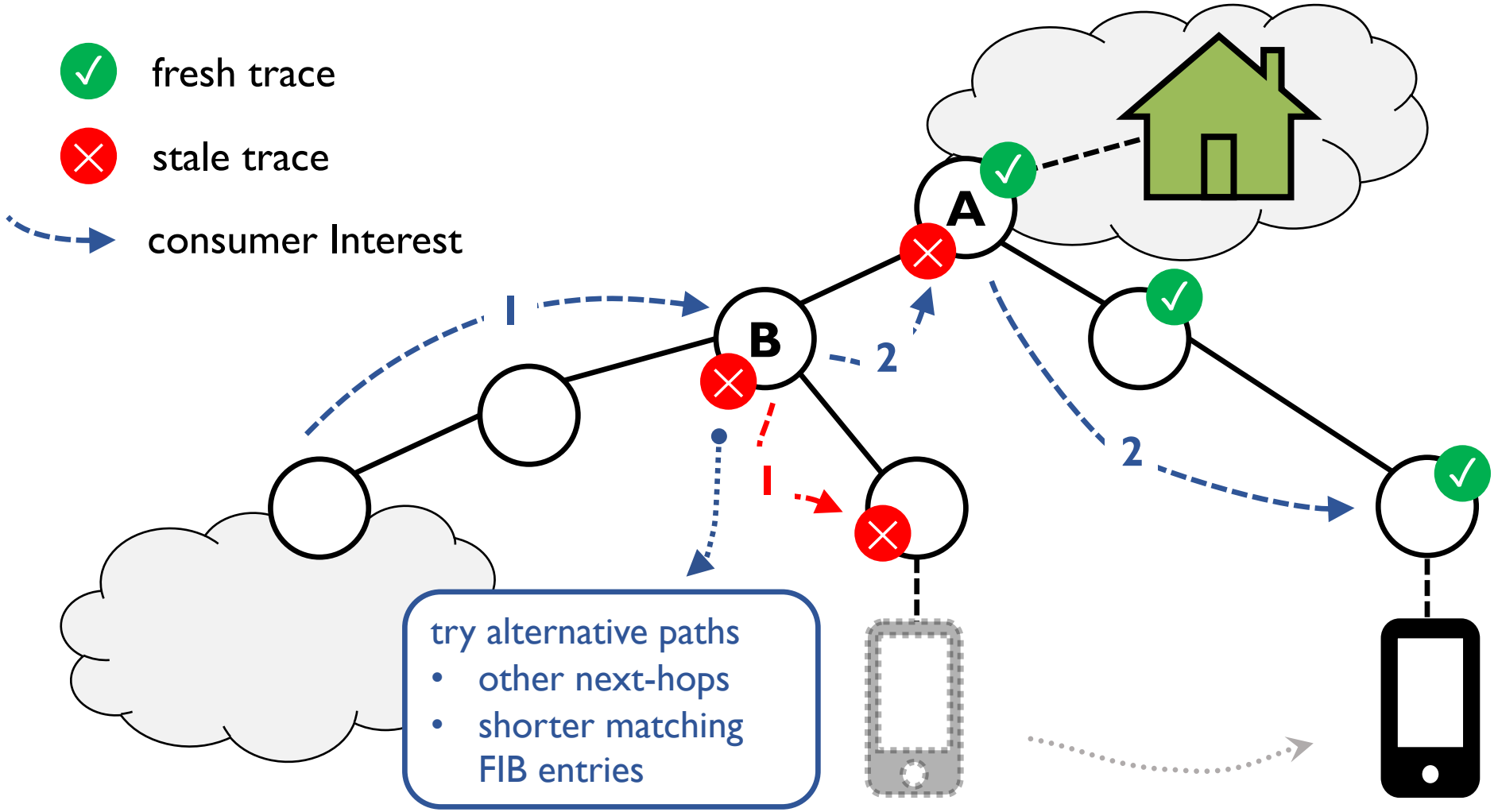


Issues with Soft-state: Stale Trace

- Stale trace emerges if previous trace is still alive **after** relocation
  - leads consumer Interests into a dead-end
- In certain cases, consumer can't reach the MP until stale trace expires
  - **path shortcut** + **stale trace** + **no fresh trace** on the forwarding path



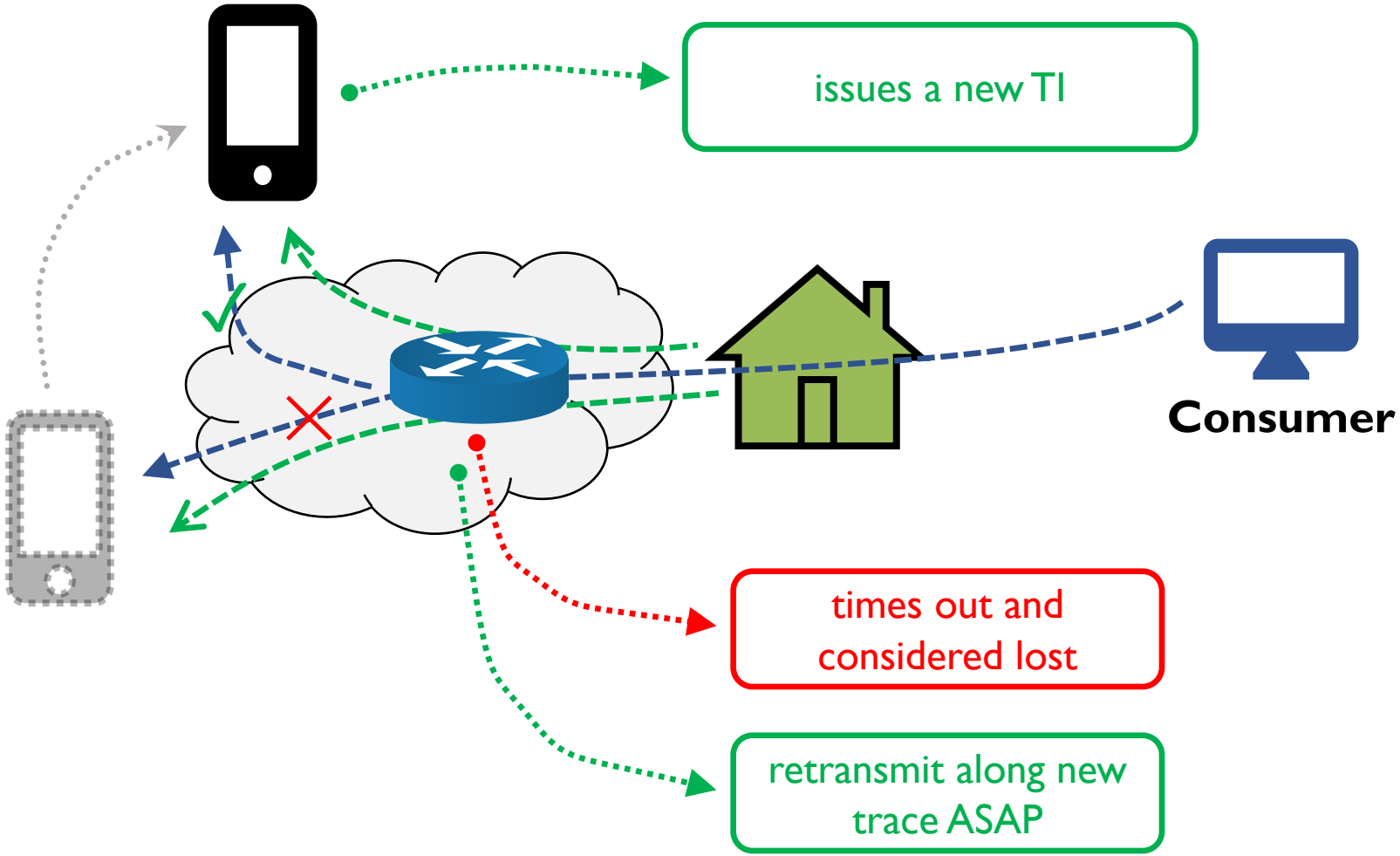
Mitigate the Impact of Stale Trace with Forwarding Strategy



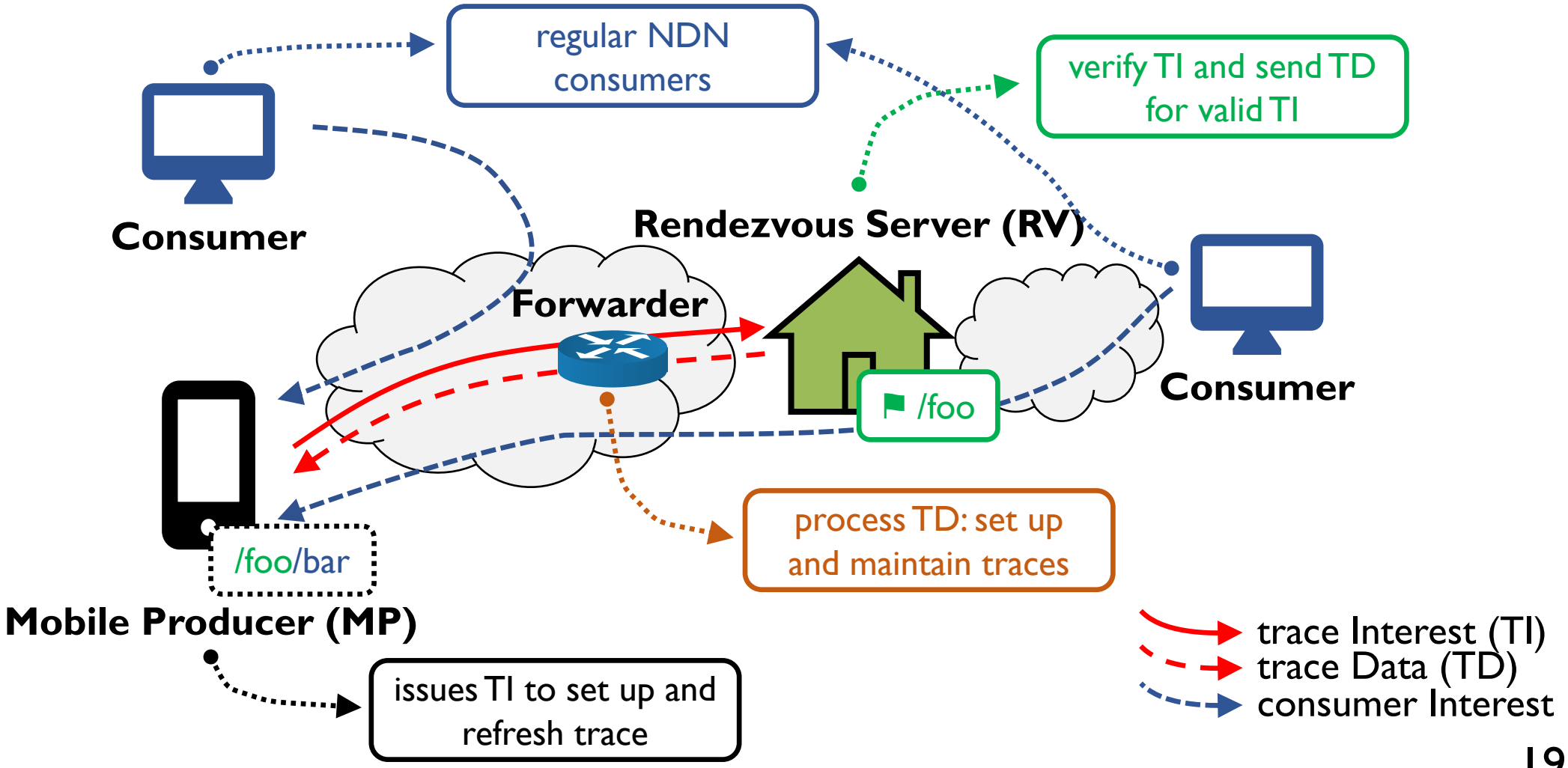
Proof of Reachability

- Upon receipt of TD, forwarding paths to the MP exists for any consumer
  - as long as the **routing prefix** is globally reachable
- Consumer Interests can always reach the MP as long as valid forwarding paths exist
  - assume that last-hop forwarder will send NACK back for failed recovery
  - the introduced forwarding strategy does a depth-first search on the forwarding tree consisting all alive traces (fresh and stale)

In-network Interest Retransmission

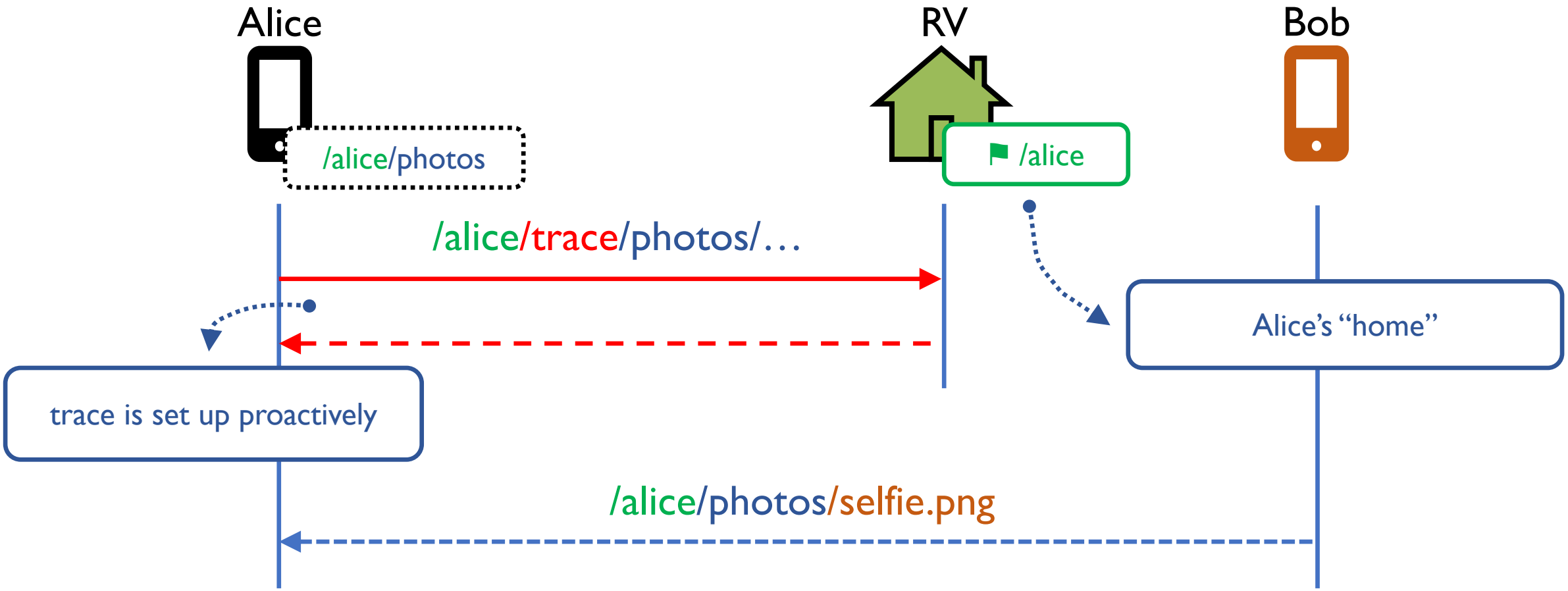


Summary

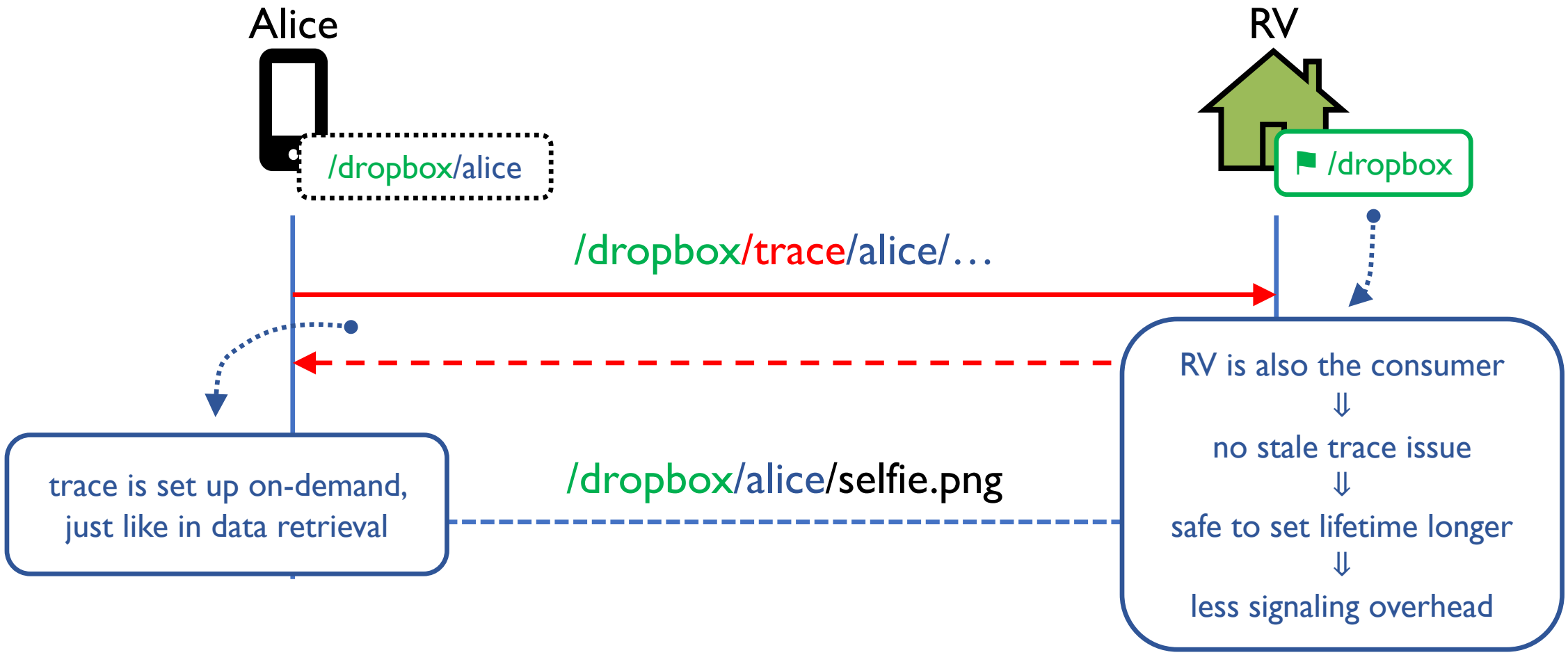

regular NDN consumers
verify TI and send TD for valid TI
Consumer
Rendezvous Server (RV)
Forwarder
Consumer
/foo
process TD: set up and maintain traces
/foo/bar
Mobile Producer (MP)
issues TI to set up and refresh trace
trace Interest (TI)
trace Data (TD)
consumer Interest

KITE and Applications

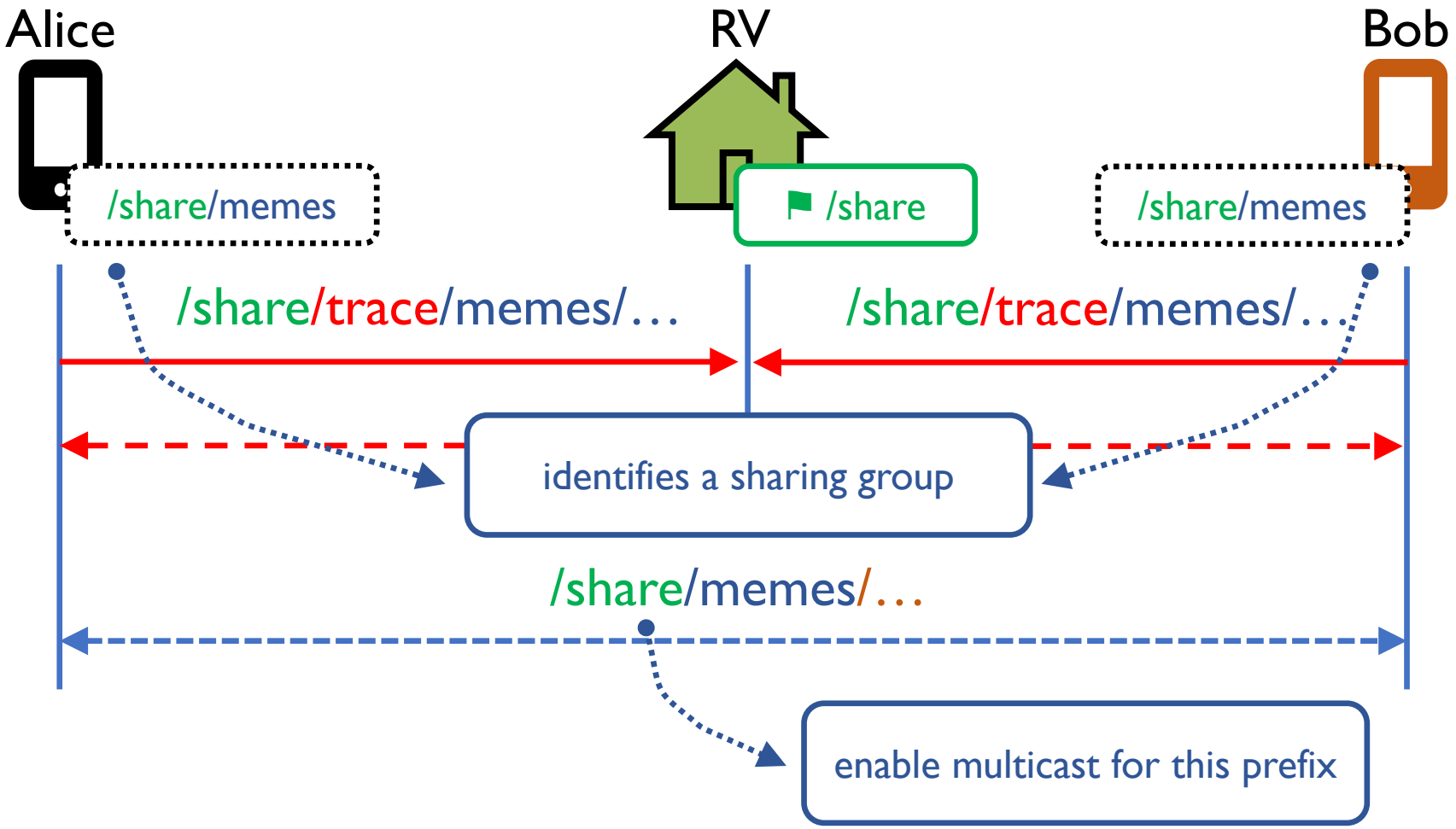
Pull


Alice
RV
Bob
/alice/photos
/alice
/alice/trace/photos/…
Alice's "home"
trace is set up proactively
/alice/photos/selfie.png

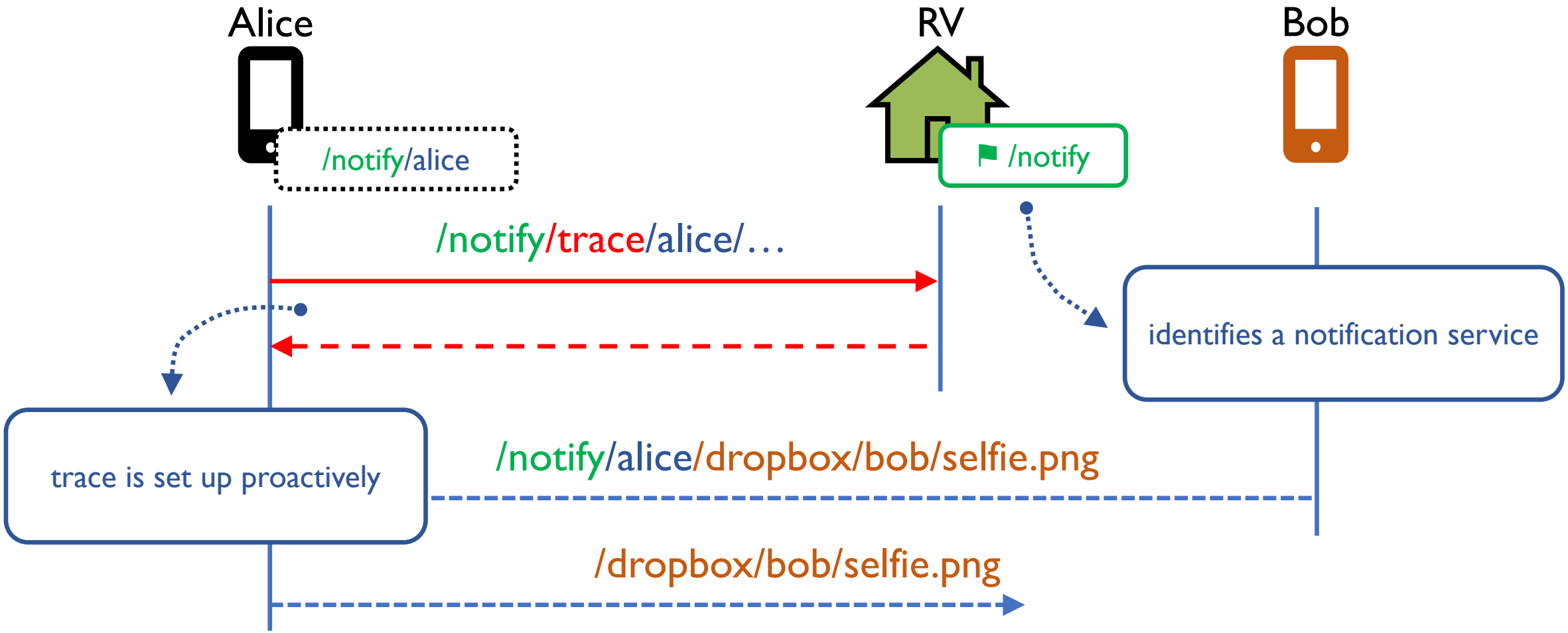
Upload


Alice
/dropbox/alice
RV
/dropbox
/dropbox/trace/alice/…
trace is set up on-demand,
just like in data retrieval
/dropbox/alice/selfie.png
RV is also the consumer
⇓
no stale trace issue
⇓
safe to set lifetime longer
⇓
less signaling overhead

Share


Alice
RV
Bob
/share/memes
/share
/share/memes
/share/trace/memes/…
/share/trace/memes/…
identifies a sharing group
/share/memes/…
enable multicast for this prefix

Push


Alice
RV
Bob
/notify/alice
/notify
/notify/trace/alice/…
identifies a notification service
trace is set up proactively
/notify/alice/dropbox/bob/selfie.png
/dropbox/bob/selfie.png

KITE vs. Other Solutions

MP-Chasing: locate the MP

- Mapping-based: data prefix is mapped to a topology-dependent locator
- Routing-based: updates the forwarding tree built with routing
  - scalability concerns
- Tracing-based: trace the MP with the stateful forwarding plane
  - ensure reachability with a routable prefix
- KITE: a tracing-based approach
  - locator-free
  - transparent to routing and data retreival
  - abuse-proof

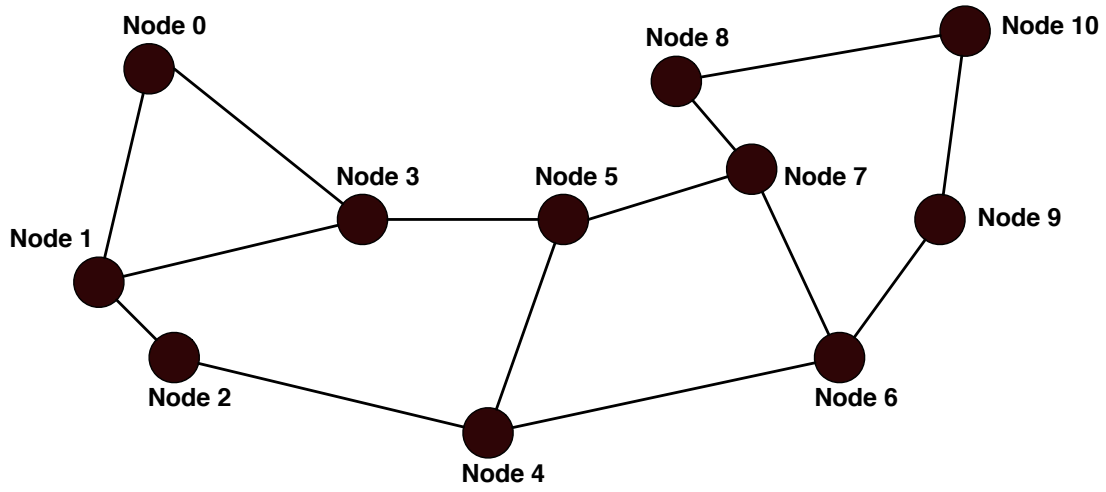
Implementation and Evaluation

Implementation

- Proof-of-concept prototype
  - open source: https://github.com/KITE-2018
  - based on "real" NDN code (NDN Forwarding Daemon, NFD)
  - two application scenarios
  - directly usable for simulations with ndnSIM 2.x

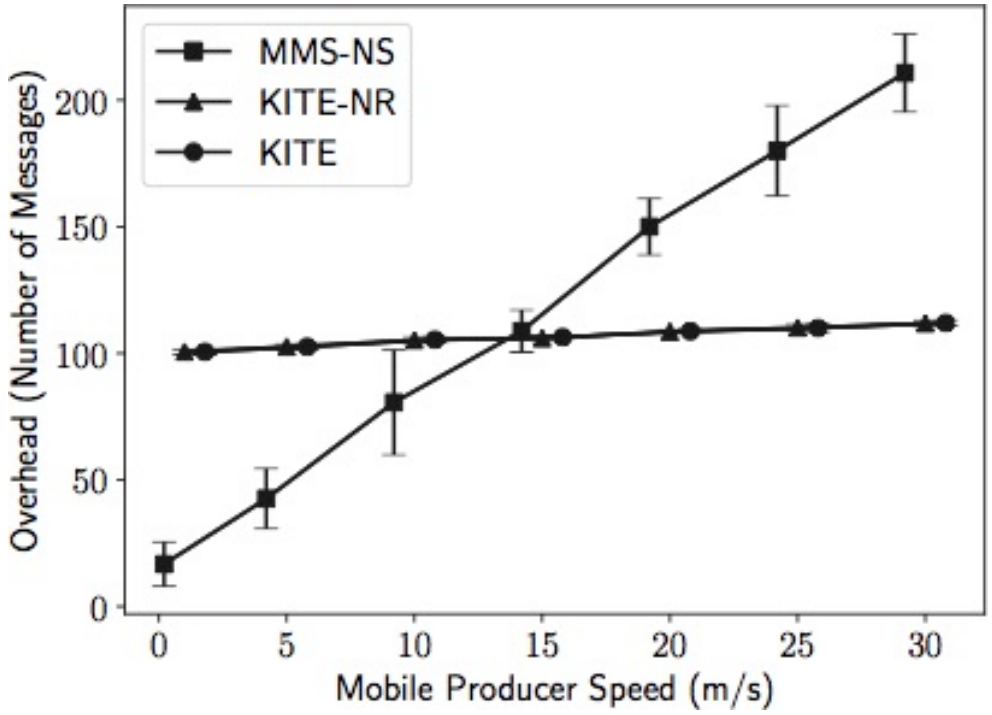
Evaluation

- KITE vs mapping-based solutions
  - simulations on a toy topology
  - Pull and Upload scenario
  - different mobility patterns
- Preliminary results, not for showcasing performance

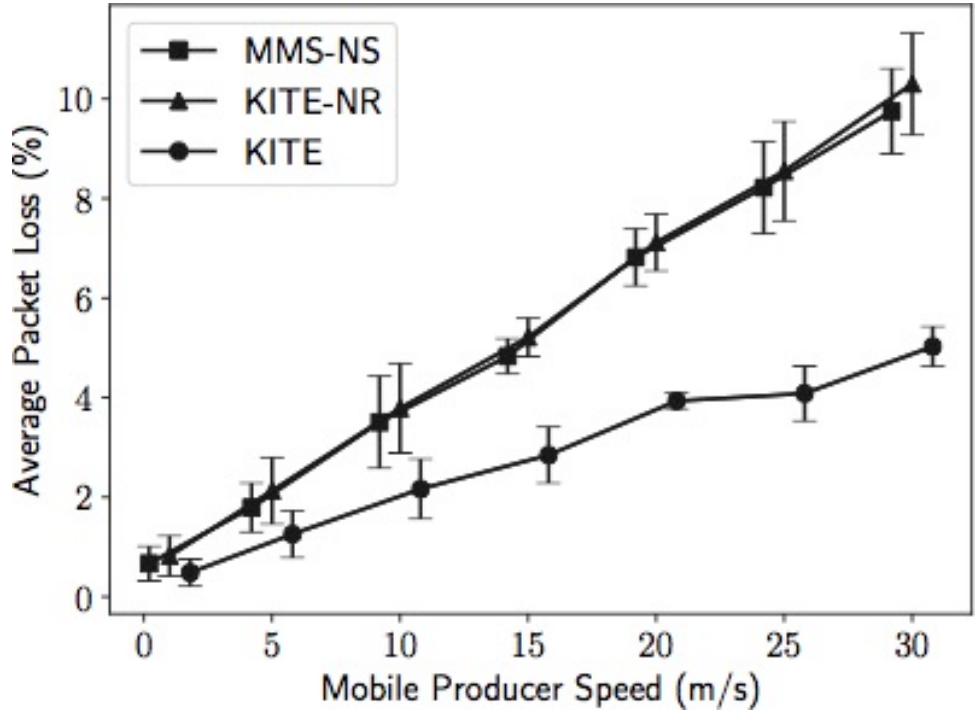


Simulation Results

- KITE is insensitive to mobility pattern



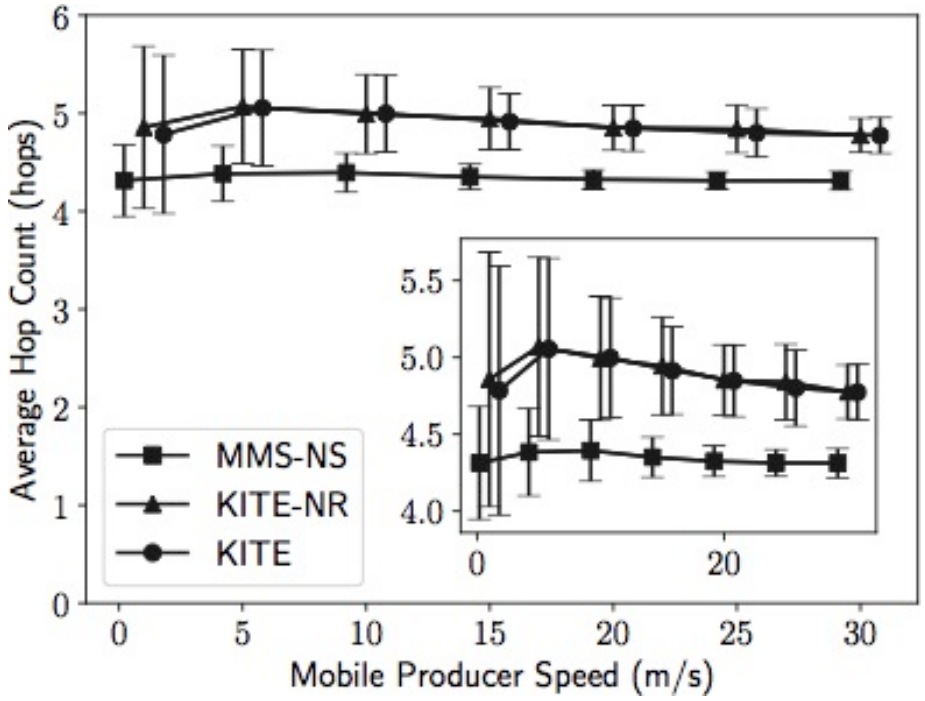
Signaling overhead - Pull



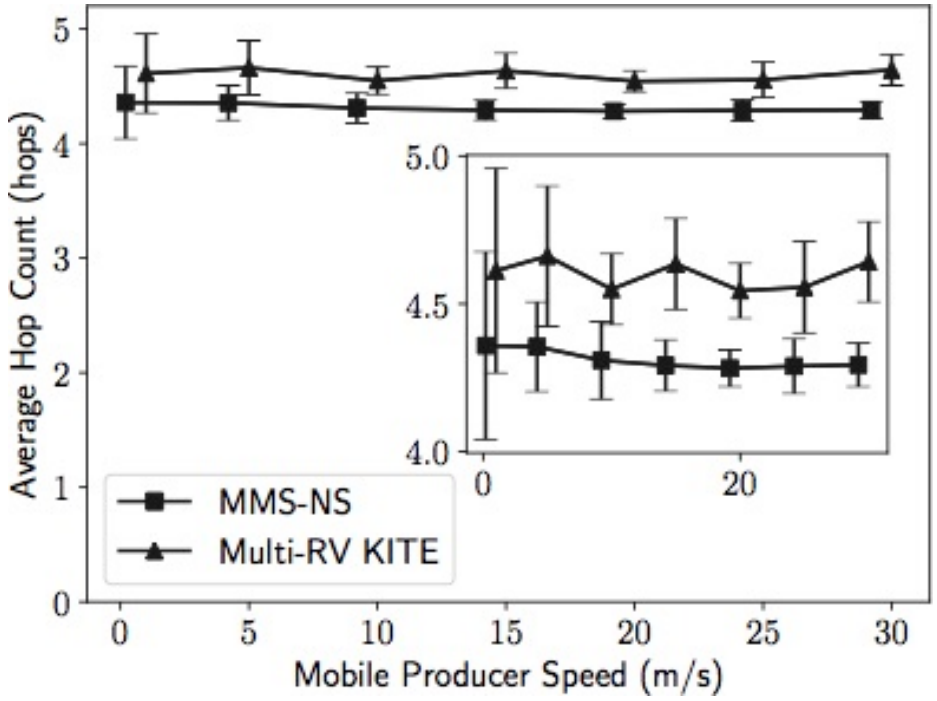
Packet loss - Pull

Simulation Results

- KITE only slightly suffers from triangle routing



Hop count - Pull



Hop count - Pull - distributed RV

Conclusion

Conclusion

- Transparency
  - to data retrieval process: no locator, no change to names
  - to routing: operates on non-routable prefixes
- Security
  - provides usable security with RV doing the verification
- Scalability
  - orthogonal to routing scalability
  - the RV functionality can be distributed across cooperative RV instances to scale with increasing number of mobile devices/prefixes/events
- Architectural impact
  - only extends the stateful forwarding plane
  - no change to packet format

Future Work

Future Work

- Efficient soft-state management
- Extensive evaluation to quantify performance in more, finer tuned simulation setups
- Integrate KITE into NDN code release
  - NFD release
  - API library release (ndn-cxx)
- Conduct experiments on NDN testbed
- KITE with distributed RV for better scalability and robustness

Thank you!

Q&A

xiazhongda@hit.edu.cn