Introduction to Sync

milcom

Military Communications for the 21st Century

LAX Marriott, Los Angeles, CA

Outline

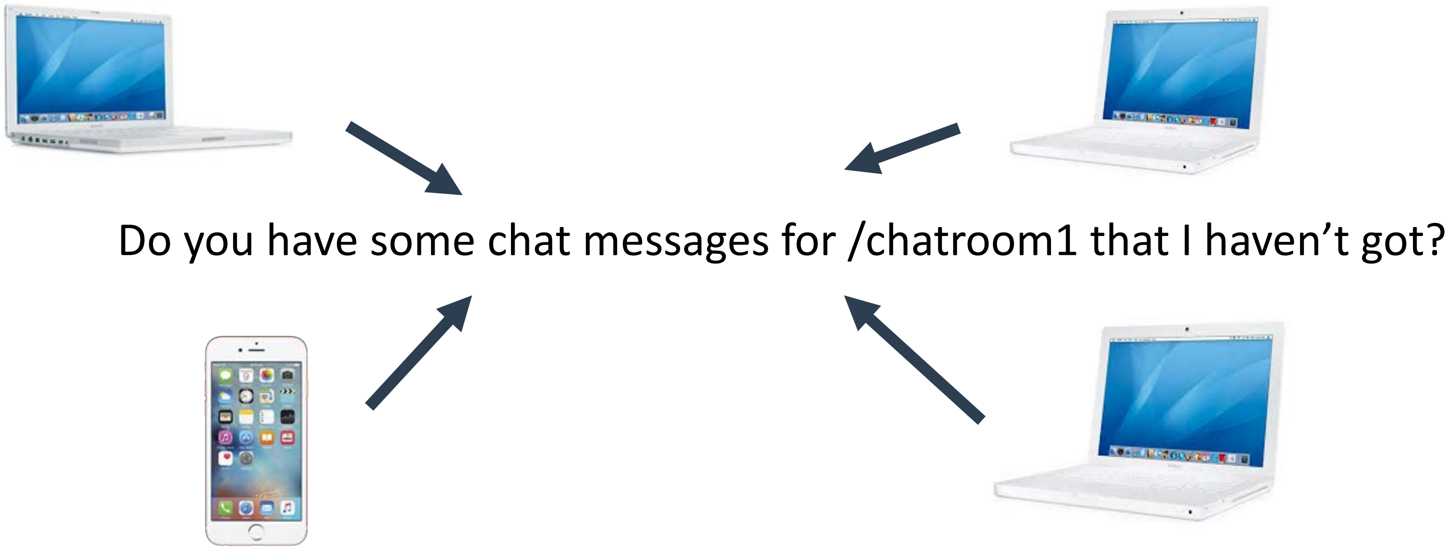
- NDN Sync overview

- Sync protocol design issues

- Sync protocol for mobile ad-hoc networks

What is NDN Sync?

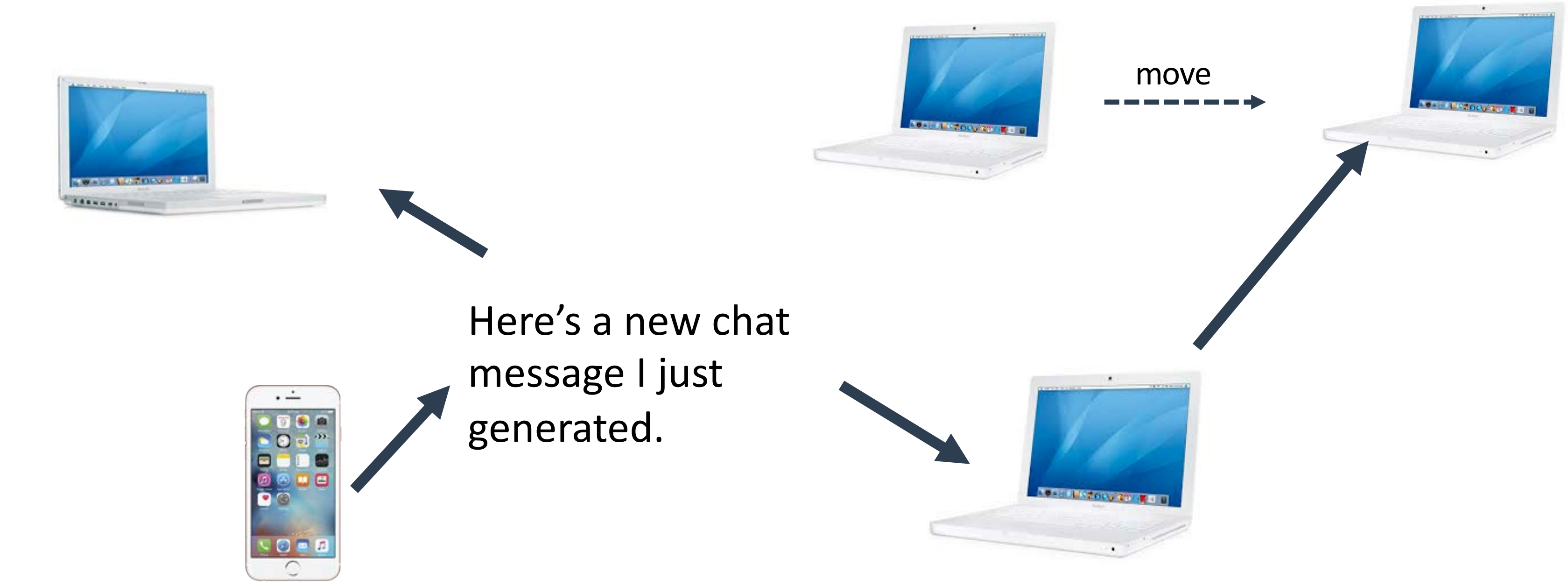
- New transport *service* for data-centric networking [1]
- Synchronizing the namespace of shared dataset for distributed applications among all participants



Do you have some chat messages for /chatroom1 that I haven’t got?

Benefits

- Supports multi-party communication
  - Applications do not need to worry about individual participant addresses or how to reach them
- Desired data can be fetched from any node.
  - Resilient delivery under mobile & intermitted connectivity

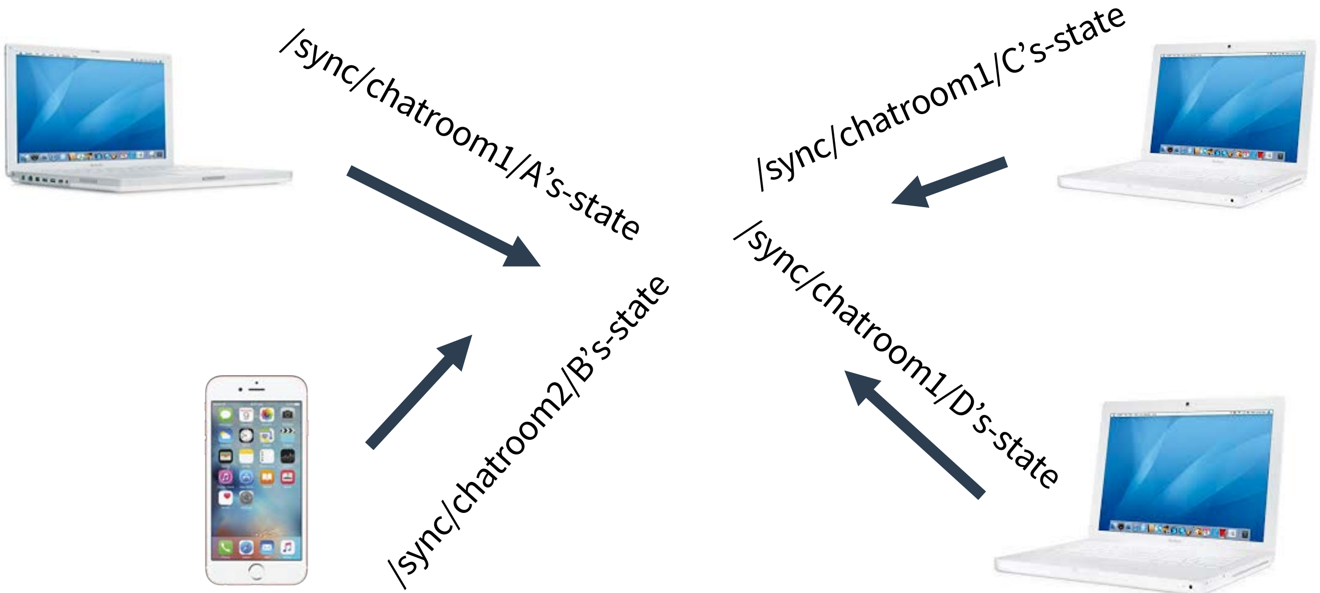


move

Here’s a new chat message I just generated.

How to sync?

- Encode the data set (names) into a compact form
- Participants exchange and compare the dataset state to detect differences



/sync/chatroom1/A's-state

/sync/chatroom2/B's-state

/sync/chatroom1/C's-state

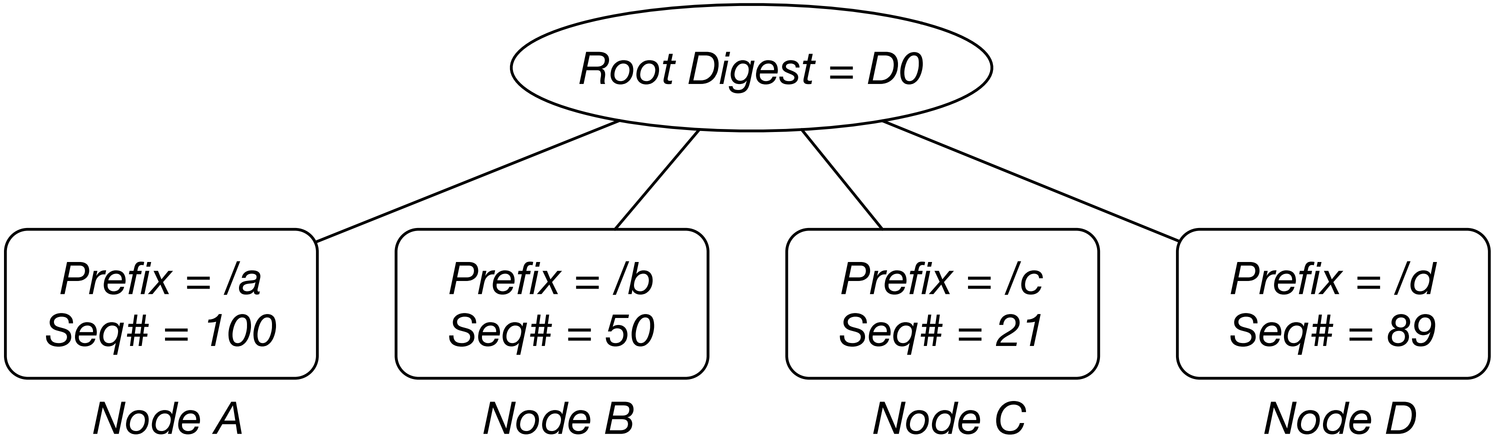
/sync/chatroom1/D's-state

Sync Design Issues

- How to represent one's dataset state?
  - to make the comparison between datasets with least delay and overhead
- Pair-wise vs. group sync
  - Pair-wise: A – B, B - C, C - D, etc.
  - Group: every node tells all other nodes its data state.
- Sync names vs. sync data
  - Sync protocol may reconcile names only.
  - Individual participants decide whether or when to retrieve missing data based on local conditions

Example: ChronoSync [3]

- Use hash as a compact form of one's dataset state
- Two-level digest tree
  - Each producer represented by <producer-name, seq#>
- Participants exchange root digest with each other to detect differences in dataset



Root Digest = D0

Prefix = /a
Seq# = 100

Node A

Prefix = /b
Seq# = 50

Node B

Prefix = /c
Seq# = 21

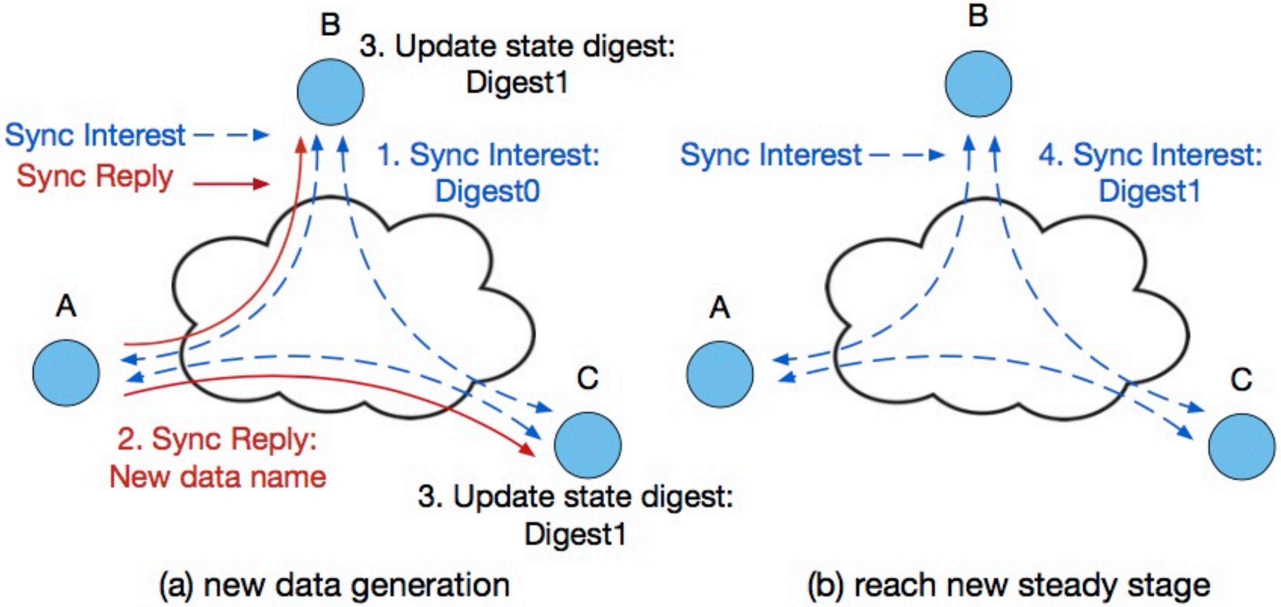
Node C

Prefix = /d
Seq# = 89

Node D

ChronoSync (cont'd)

- Multicast sync interests to all participants in an app group



(a) new data generation

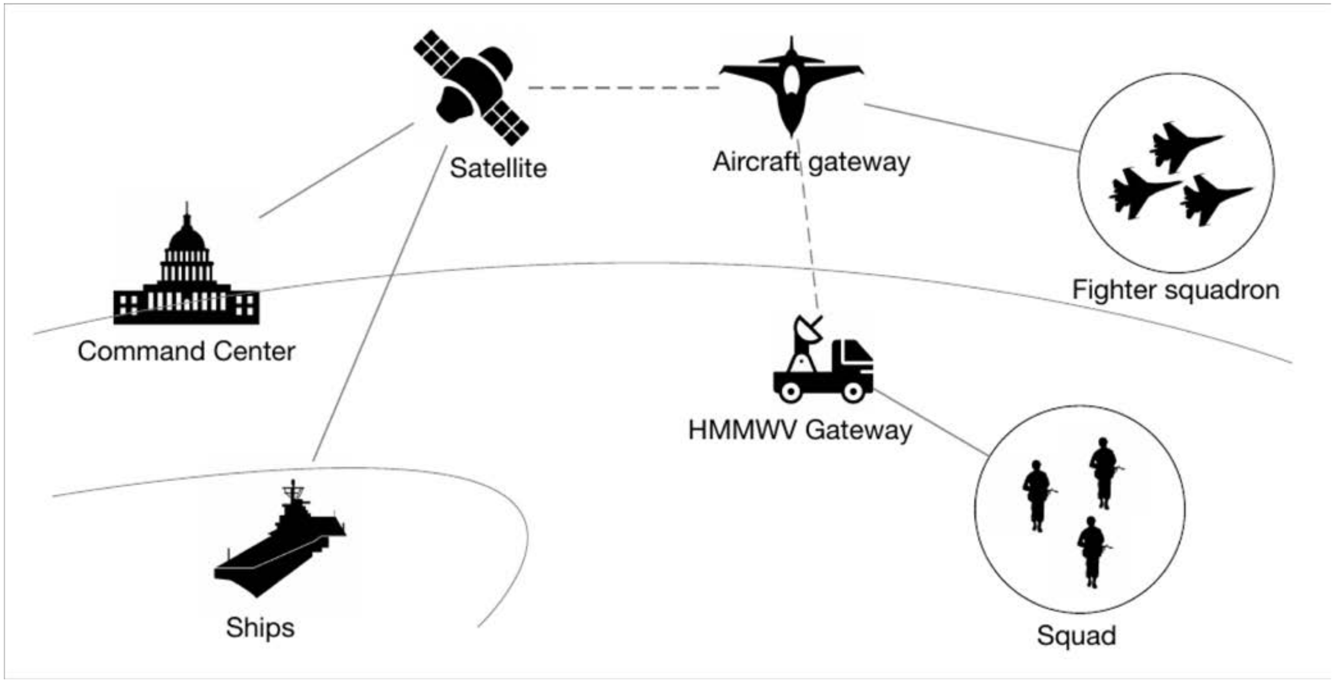
(b) reach new steady stage

Limitation: cannot directly decode dataset difference from comparing digests

- Applying a few good engineering hints can work well under normal conditions

Sync for Mobile Ad Hoc Networks

- Challenge: Intermittent connectivity with mobility
  - Short-lived connections
  - Dynamically changing topology
- Need to rethink Sync design [1]



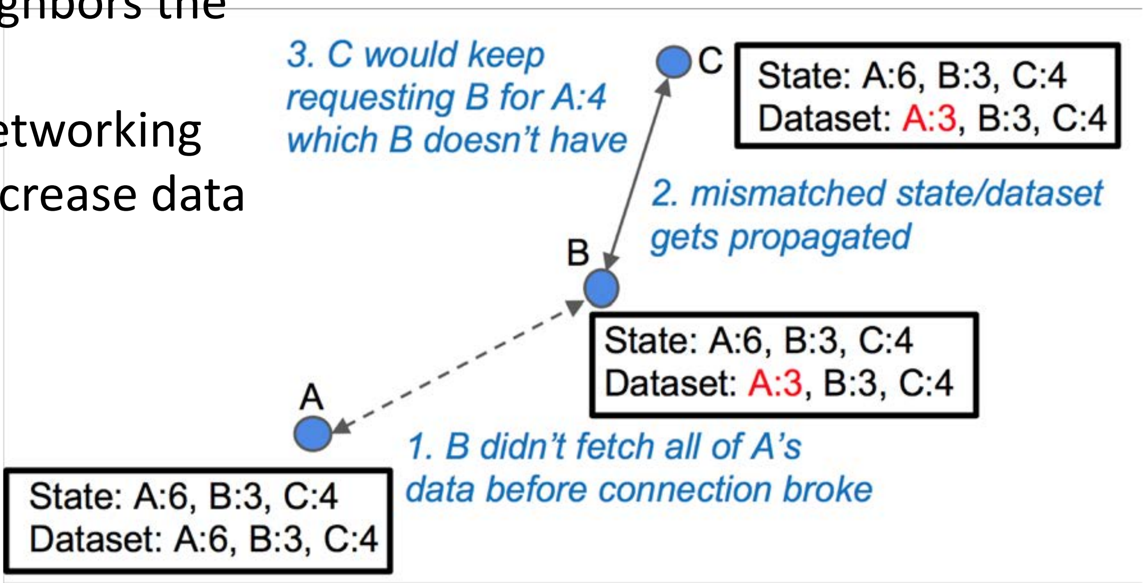
MANET Sync [2]

- State encoding
  - State vector [A:5, B:7, …]
  - Carried in Sync Interest
- State change detection is achieved directly from state vector comparison [5].
  - Whenever one generate new data: send new state vector
  - Periodic transmission of Sync Interests as backup

MANET Sync [2]

- New issues in dataset name sync versus data sync
  - Not all nodes are connected at any given time
  - Not all nodes fetch all data
  - One may learn the name of a new piece of data from neighbor which does not have the data

Solution-1: only inform neighbors the names if one has the data
Solution-2: engineer the networking with data repositories to increase data availability



State: A:6, B:3, C:4
Dataset: A:3, B:3, C:4

3. C would keep requesting B for A:4 which B doesn't have

2. mismatched state/dataset gets propagated

State: A:6, B:3, C:4
Dataset: A:3, B:3, C:4

1. B didn't fetch all of A's data before connection broke

State: A:6, B:3, C:4
Dataset: A:6, B:3, C:4

Summary

- Transport layer service for NDN: dataset name reconciliation for distributed applications
- Represent the dataset using a compact format over data names
- Exchange dataset state to detect missing data names
- Implemented in libraries

Join the NDN session Tuesday afternoon to learn more details:
“A Brief Introduction to NDN Dataset Synchronization”

Further reading

1) T. Li, W. Shang, A. Afanasyev, L. Wang, L. Zhang, “A Brief Introduction to NDN Dataset Synchronization,” in Proceedings of MILCOM 2018, Oct. 2018

2) X. Xu, H. Zhang, T. Li, L. Zhang, “Achieving Resilient Data Availability in Wireless Sensor Networks,” in Proceedings of IEEE ICC, May 2018

3)Z. Zhu and A. Afanasyev, “Let’s ChronoSync: Decentralized dataset state synchronization in Named Data Networking,” in Proceedings of IEEE ICNP, 2013.

4) W. Fu, H. Ben Abraham, and P. Crowley, “Synchronizing namespaces with invertible bloom filters,” in Proceedings of the Eleventh ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ser. ANCS ’15, 2015.

5) W. Shang, A. Afanasyev, L. Zhang, “VectorSync: Distributed Dataset Synchronization over Named Data Networking,” NDN Technical Report NDN-0056, Mar. 2018

6) M. Zhang, V. Lehman, L. Wang, "Scalable Name-based Data Synchronization for Named Data Networking," in Proceedings of the IEEE INFOCOM 2017, May 2017

7) M. Mitzenmacher, “Compressed bloom filters,” IEEE/ACM Transactions on Networking (TON), vol. 10, no. 5, pp. 604–612, 2002.