

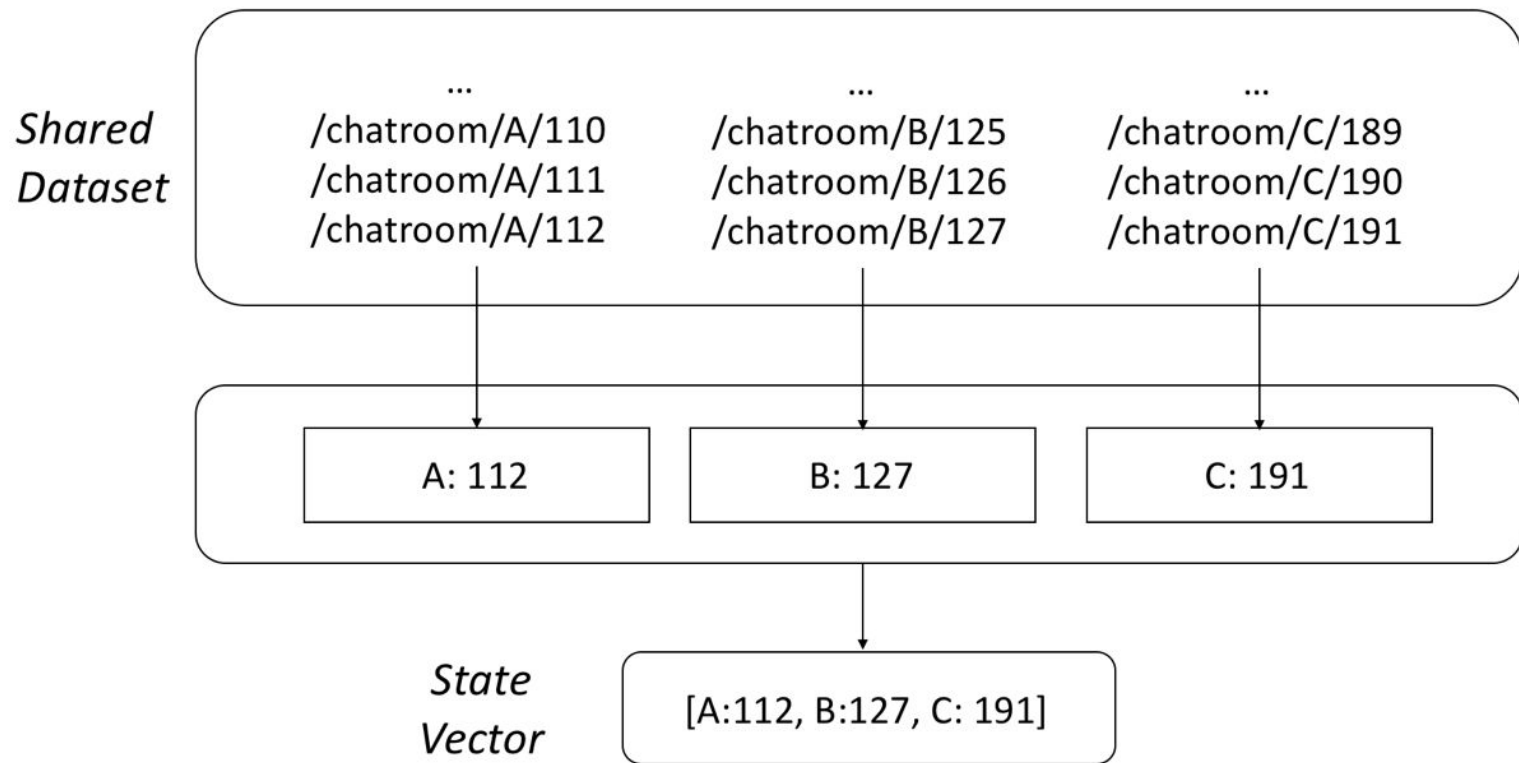
# Implement State Vector Sync (SVS)

Haitao Zhang, Saurab Dulal, Tianxiang Li, Jeff Thompson

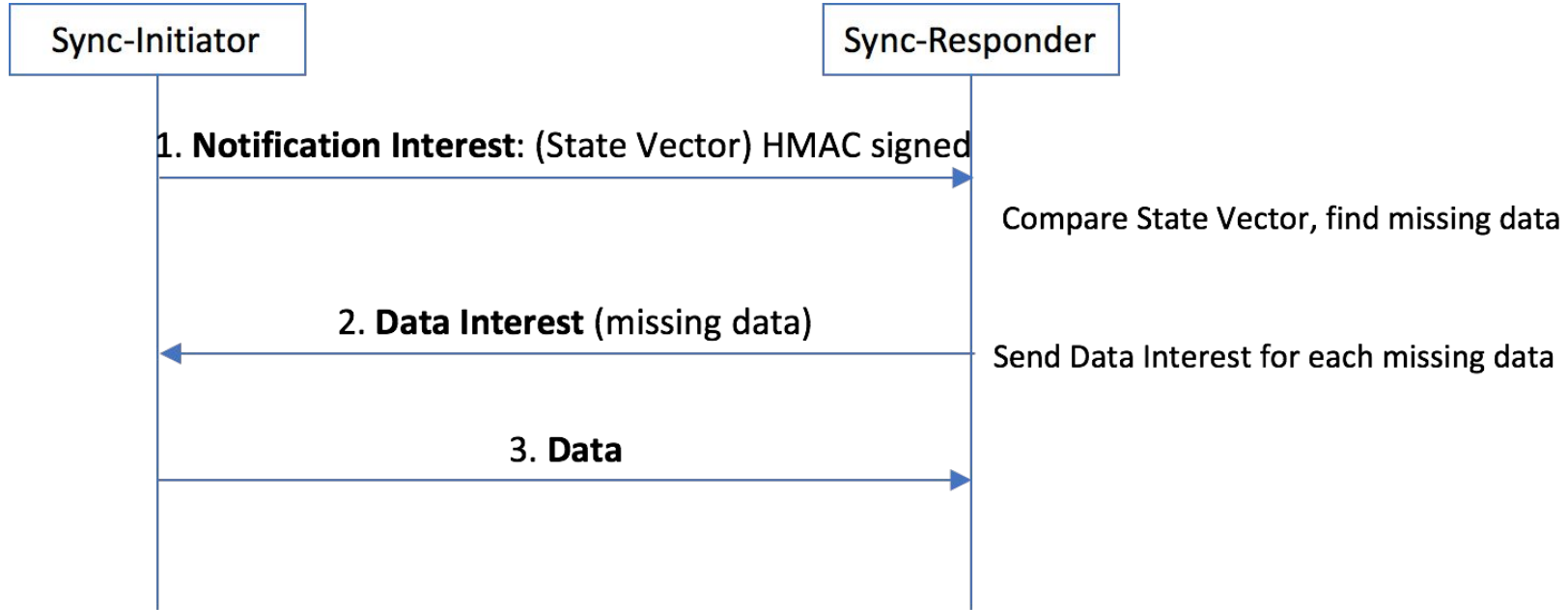
# Lessons from previous protocols

- Chronosync
  - Representing data using monotonically increasing sequence numbers
- VectorSync
  - represent the state of the shared dataset efficiently using a version vector
  - a leader-based group membership management
- State Vector Sync
  - Couples member ID and data sequence into State Vector

# State Vector



# Basic SVS Process



When to send Notification Interest: New Data triggered

# Extra Mechanisms

- Problem: A member joins late may not be able to get all the other members states
  - A two node example:
    - A is the first member in a group, it publishes 10 data packets, then stops publishing
    - B joins late, it publishes 5 data packets, then stops publishing
    - B can not get A's data packets as A never publishes another notification Interest
- Solution:
  - Each member Periodically sends notification Interest
  - When a member receives a state vector that contains an id with a lower seq#, it sends out a notification Interest (or replies with an ACK containing its state vection).

# Remaining Problems

- What if the state vector grows too large? I.e., there are too many members in the group
- What if we need to encrypt the state vector?

# Implementation

- Implemented in Python
- Use the standard APIs defined by Jeff T for all other Sync Protocols
- Tested using a chat applicaiton: SVS chat

# Screenshot of the SVS-chat

```
16:24 $ python test_svs_chat.py
Enter your chat username: haitao
Enter your hub prefix [ndn/edu/ucla/remap]: /ndn/edu/ucla/cs
Enter the chatroom name [ndnchat]:
Connecting to localhost, Chatroom: ndnchat, Username: haitao

Enter your chat message. To quit, enter "leave" or "exit".
Register prefix succeeded with the NFD forwarder for prefix /ndn/broadcast/SvsChat/ndnchat
Member: haitao
Register prefix succeeded with the NFD forwarder for prefix /ndn/edu/ucla/cs/ndnchat/asBbHo0Znh
Received broadcast state vector {'/ndn/edu/ucla/remap/ndnchat/Byjha92NnD': 1}
jeff: Hi from Jeff.
hi there
Broadcast new seq # 0. State vector {'/ndn/edu/ucla/cs/ndnchat/asBbHo0Znh': 0, '/ndn/edu/ucla/remap/ndnchat/Byjha92NnD': 1}
haitao: hi there
Received broadcast state vector {'/ndn/edu/ucla/cs/ndnchat/asBbHo0Znh': 0, '/ndn/edu/ucla/remap/ndnchat/Byjha92NnD': 2}
jeff: It works.
Received broadcast state vector {'/ndn/edu/ucla/cs/ndnchat/asBbHo0Znh': 0, '/ndn/edu/ucla/remap/ndnchat/Byjha92NnD': 3}
Broadcast new seq # 1. State vector {'/ndn/edu/ucla/cs/ndnchat/asBbHo0Znh': 1, '/ndn/edu/ucla/remap/ndnchat/Byjha92NnD': 3}
Broadcast new seq # 2. State vector {'/ndn/edu/ucla/cs/ndnchat/asBbHo0Znh': 2, '/ndn/edu/ucla/remap/ndnchat/Byjha92NnD': 3}
jeff: Leave
Broadcast new seq # 3. State vector {'/ndn/edu/ucla/cs/ndnchat/asBbHo0Znh': 3, '/ndn/edu/ucla/remap/ndnchat/Byjha92NnD': 3}
```