ChronoChat: a Server-less Multi-User Instant Message Application Over NDN

Zhenkai Zhu, Alexander Afanasyev, Yingdi Yu, Lixia Zhang

Multi-user applications are commonly implemented using a centralized server. How to exchange chat messages without the centralized server?

With NDN, we fetch chat messages directly rather than connecting to server.

A chatroom consists of multiple users.

A data set is the union of all chat messages produced by users.

By naming chat messages sequentially, a user’s message subset can be represented as (name, prefix, max(seqNo)).

The state, or knowledge, of the data set consists of such pairs.

State reconciliation

One may receive inconsistent states:
- Network delay/Packet loss
- Simultaneous data generation
- Network partition

Reconcile state:
- keep a change log to identify old digest
- send missing changes to reconcile
- remove unrecognized digest
- send current state to recover

ChronoSync to ChronoChat

- Naming convention:
  - chatroom:/ndn/multicast/ChronoChat/[chatroomName]
  - sync interest: /<chatroomName>/[digest]
  - user prefix: /<userNamespace>/ChronoChat/[chatroomName]
  - chat msg: /<userPrefix>/[sessionID]/[seqNo]

- Heartbeat message:
  - a special chat message
  - automatically sent when user is idle
  - 1 heartbeat per minute

- Fetching strategy:
  - always fetch each user’s latest chat message

State of a data set

A chatroom consists of multiple users.

A data set is the union of all chat messages produced by users.

By naming chat messages sequentially, a user’s message subset can be represented as (name, prefix, max(seqNo)).

The state, or knowledge, of the data set consists of such pairs.

Security consideration

- Authenticate user membership:
  - web-of-trust: through endorsements of existing users
  - new users join a chatroom by invitations

- Authenticate user identity:
  - hierarchical naming hierarchy
  - web-of-trust: 3 levels of endorsement

- Users are trusted for their own state updates and chat messages

- Users are authenticated against their own integrity

- Implementations:
  - Open source native app in Mac OS X and Linux
  - https://github.com/named-data/ChronoChat
  - Binaries also available

ChronoSync: efficient state synchronization

- Represent state as a digest tree:
  - The root digest summarizes the state of the whole set
  - Each child node summarizes the state of the whole set

- Two packet types:
  - Every piece of data has a name
  - Security is built into data

- Two packet types:
  - Every piece of data has a name
  - Security is built into data

- Periodically reconstruct state & change log:
  - initiated by reset interest
  - all users clean up state & change log
  - active users add themselves back again

Scalable maintenance

- Keep a scalable state & change log:
  - remove inactive users
  - remove old change log

- Periodically reconstruct state & change log:
  - initiated by reset interest
  - all users clean up state & change log
  - active users add themselves back again

NDN: a new Internet architecture

- Receiver-driven communications:
  - Send Interest to retrieve Data
  - One Interest brings at most one Data packet

- Intelligent data plane:
  - Router maintain “Pending Interest Table”
  - Aggregation of Interests by routers
  - Natural support for data multicast

- Two packet types:
  - Every piece of data has a name
  - Security is built into data

- Intelligent data plane:
  - Router maintain “Pending Interest Table”
  - Aggregation of Interests by routers
  - Natural support for data multicast

- State of a data set:
  - A chatroom consists of multiple users
  - A data set is the union of all chat messages produced by users

- By naming chat messages sequentially, a user’s message subset can be represented as (name, prefix, max(seqNo))

- The state, or knowledge, of the data set consists of such pairs.

- Scalable maintenance:
  - Keep a scalable state & change log:
    - remove inactive users
    - remove old change log

- ChronoSync: efficient state synchronization:
  - Represent state as a digest tree:
    - The root digest summarizes the state of the whole set
    - Each child node summarizes the state of the whole set

- Two packet types:
  - Every piece of data has a name
  - Security is built into data

- Periodically reconstruct state & change log:
  - initiated by reset interest
  - all users clean up state & change log
  - active users add themselves back again

- ChronoSync: efficient state synchronization:
  - Represent state as a digest tree:
    - The root digest summarizes the state of the whole set
    - Each child node summarizes the state of the whole set

- Two packet types:
  - Every piece of data has a name
  - Security is built into data

- Periodically reconstruct state & change log:
  - initiated by reset interest
  - all users clean up state & change log
  - active users add themselves back again

- ChronoSync: efficient state synchronization:
  - Represent state as a digest tree:
    - The root digest summarizes the state of the whole set
    - Each child node summarizes the state of the whole set

- Two packet types:
  - Every piece of data has a name
  - Security is built into data

- Periodically reconstruct state & change log:
  - initiated by reset interest
  - all users clean up state & change log
  - active users add themselves back again

- ChronoSync to ChronoChat

- Naming convention:
  - chatroom:/ndn/multicast/ChronoChat/[chatroomName]
  - sync interest: /<chatroomName>/[digest]
  - user prefix: /<userNamespace>/ChronoChat/[chatroomName]
  - chat msg: /<userPrefix>/[sessionID]/[seqNo]

- Heartbeat message:
  - a special chat message
  - automatically sent when user is idle
  - 1 heartbeat per minute

- Fetching strategy:
  - always fetch each user’s latest chat message

This material is based upon work supported by the National Science Foundation under contract No. CNS-1040868.