b
UNIVERSITÄT
BERN

Synchome

Evaluation of Synchronization Application for Mobile Content Retrieval

Nina Mujkanovic
Computer Networks and Distributed Systems
University of Berne

16.03.2015

Roadmap

> What is CCNx

> CCNx synchronization protocol

> Problem statement

> Synchome design

> Evaluation

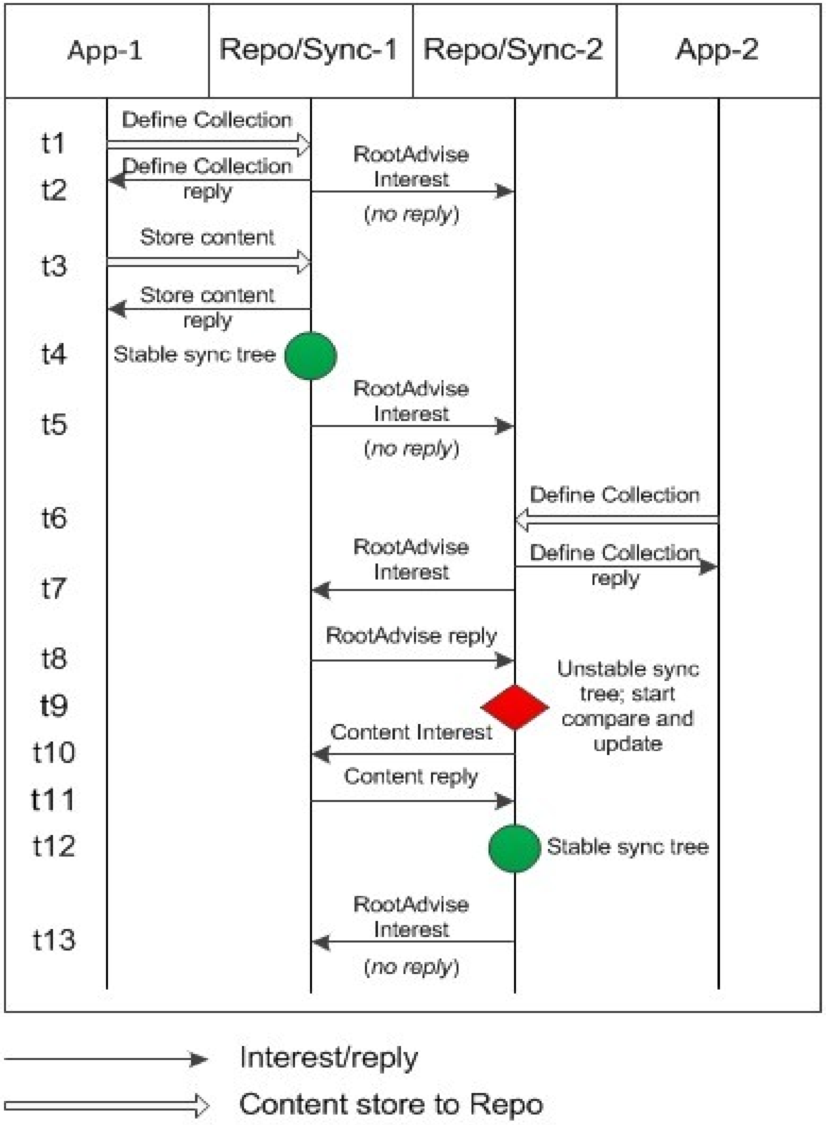
What is CCNx

- CCN = Content Centered Networking
- Communication based on names
- Everything is a message – Interest and Data messages
- CCNx = open source implementation of CCN

CCNx synchronization protocol

- Keeps data automatically synchronized
- Data is stored in persistent storage - repositories
- User defines what to synchronize
- New content is synchronized between participating hosts

CCNx synchronization protocol



Problem statement

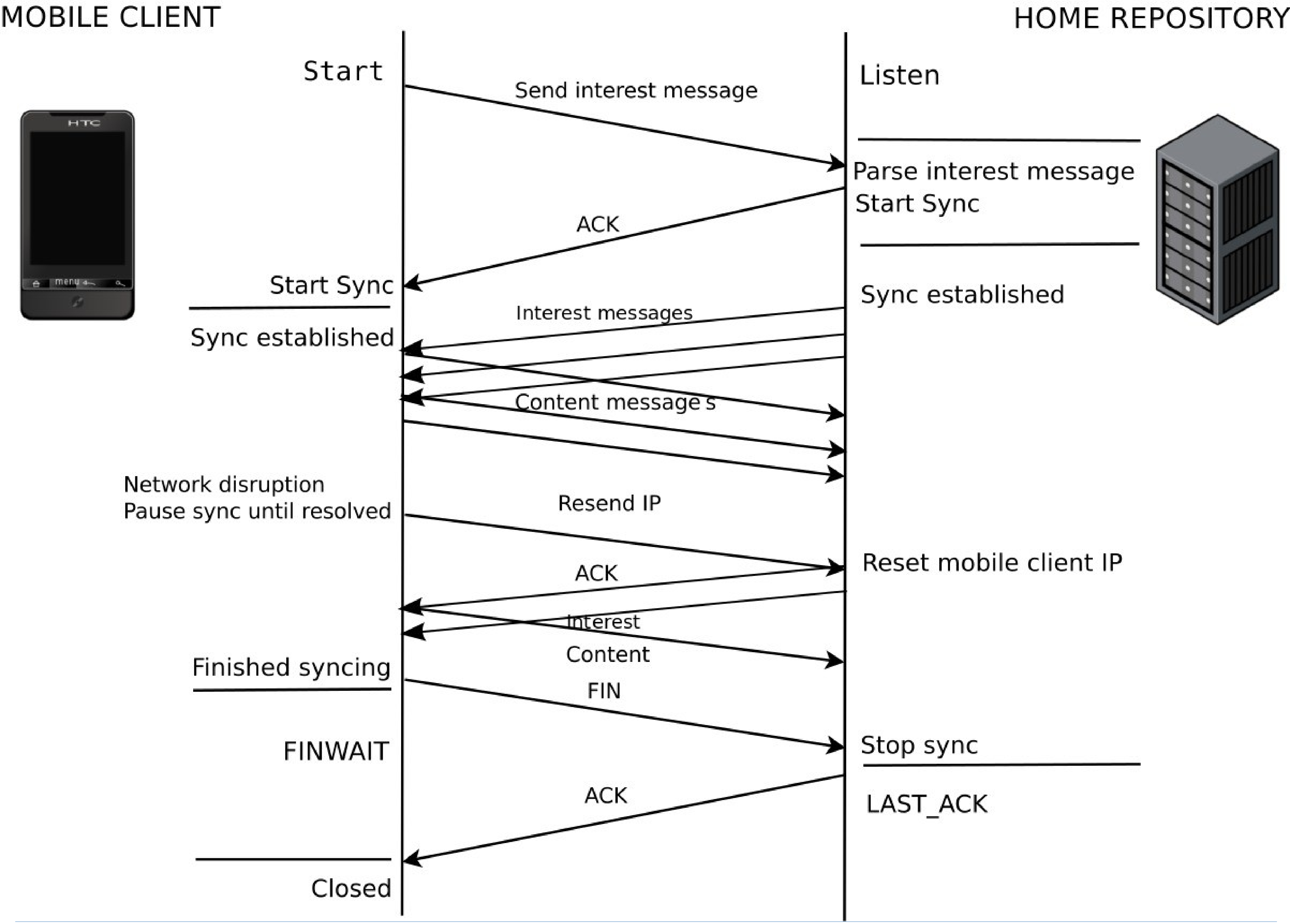
- Synchronizing hosts must be in same local subnet or connected by tunnel → additional overhead for mobile clients
- Synchronization runs infinitely unless aborted by user → continued energy expenditure
- Synchronization protocol continues exchanging messages when not actively synchronizing content → inefficient for mobile clients

Synchome

- Synchronization protocol for mobile clients
- Registers hosts IP addresses as locators in CCNx routing tables
- Detects synchronization disrupts (e.g. no AP in vicinity) and re-registers hosts if necessary
- Exits once synchronization of content is finished

Synchome

- Consists of:
- Client application (registers server, sends initialization Interest to server, runs monitoring application, exits once finished succesfully)
- Server application (home repository, constantly running and listening for Interests)
- Synchronization monitoring application



MOBILE CLIENT

HOME REPOSITORY

Start

Listen

Send interest message

Parse interest message
Start Sync

ACK

Start Sync

Sync established

Interest messages

Sync established

Content message s

Network disruption
Pause sync until resolved

Resend IP

Reset mobile client IP

ACK

Interest

Content

Finished syncing

FIN

FINWAIT

Stop sync

ACK

LAST_ACK

Closed

Evaluation

> synchome vs. ccnsyncslice (implementation of synchronization protocol)

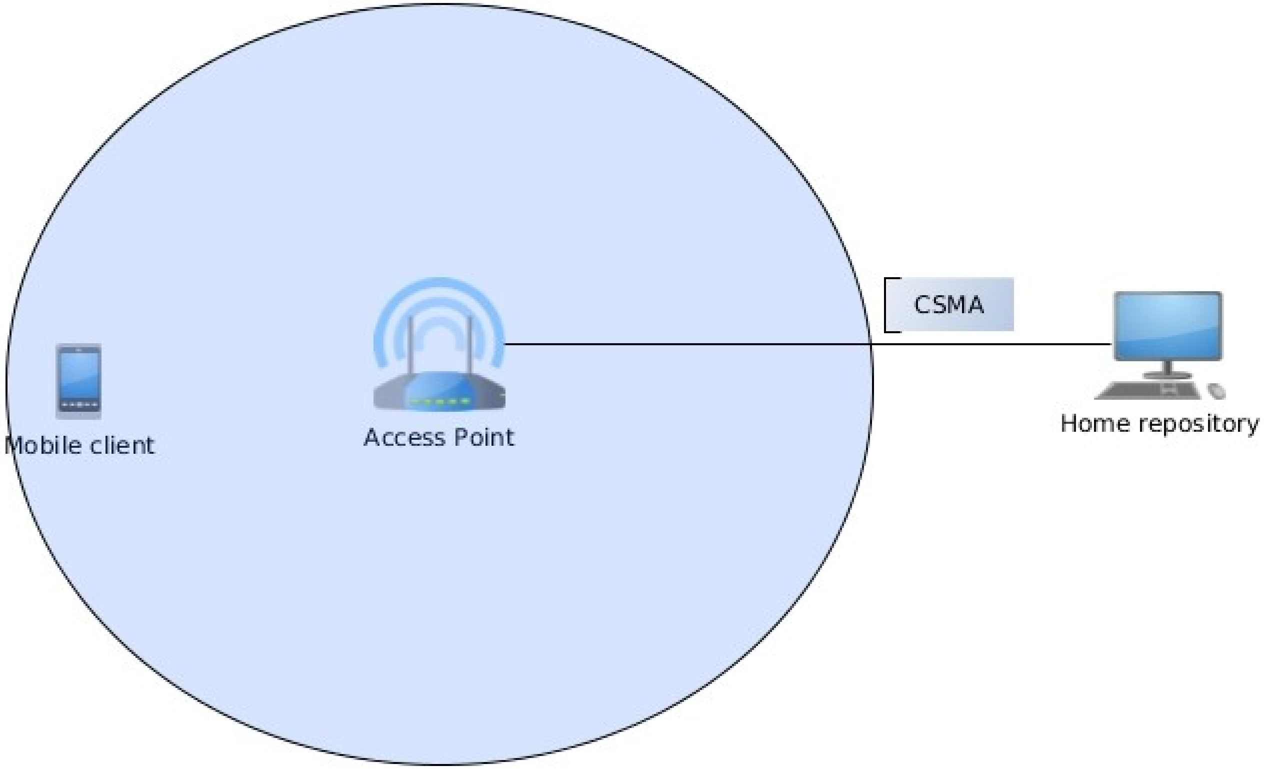
> Synchronization of content with/without periods where no new content is available

> Measure synchronization time

> Use CCNx log to analyze Interest and Data messages passed between client and host

Evaluation

> Evaluated on UBELIX using DCE-ns3



CCNx synchronization protocol

- Keeps information stored in synchronization tree. Parent folder hash is tree roothash, subfolder hashes are child nodes, etc
- Rootadvise carrying roothash sent every 20 seconds. If roothashes differ, new content available
- When synchronizing, rootadvise Interests are exchanged every second
- Node fetch Interests carrying child node hashes exchanged simultaneously with content until synchronization finished

CCNx synchronization protocol

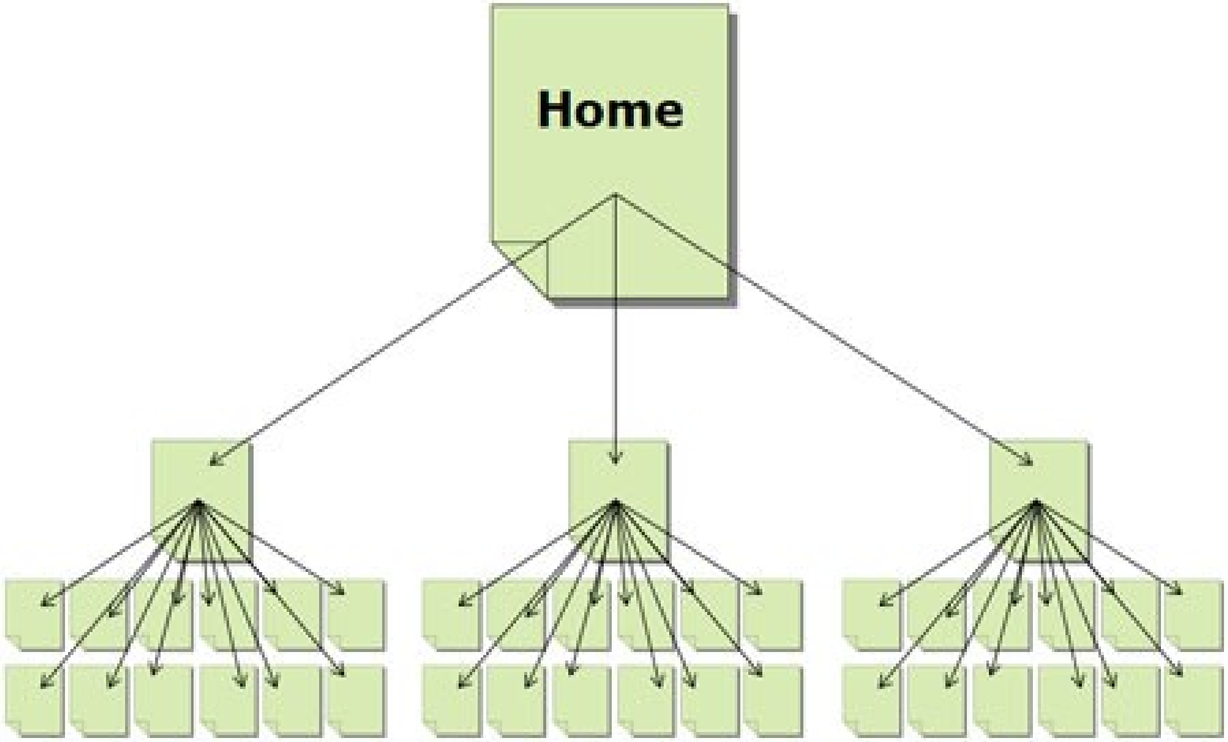
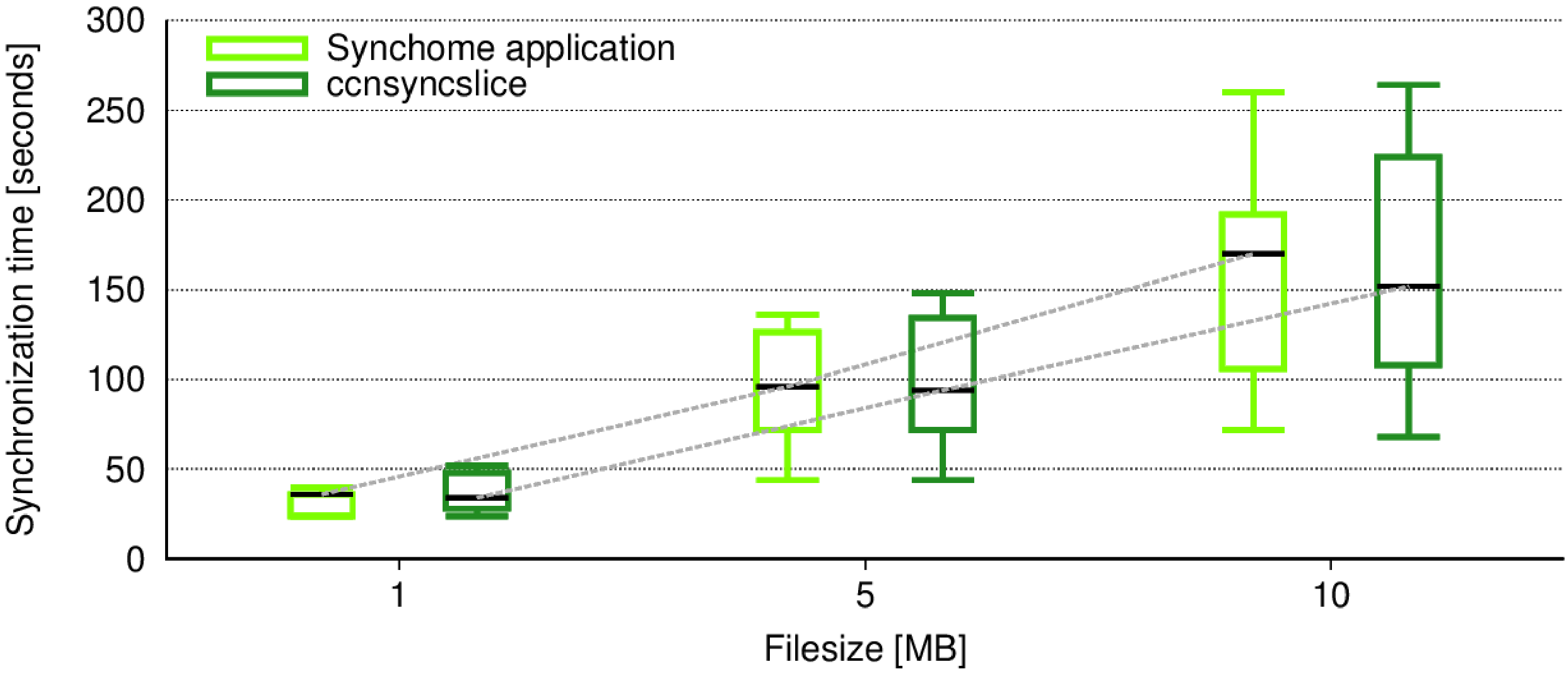


Image courtesy of http://static.searchengineguide.com/images/directory-ideal.jpg

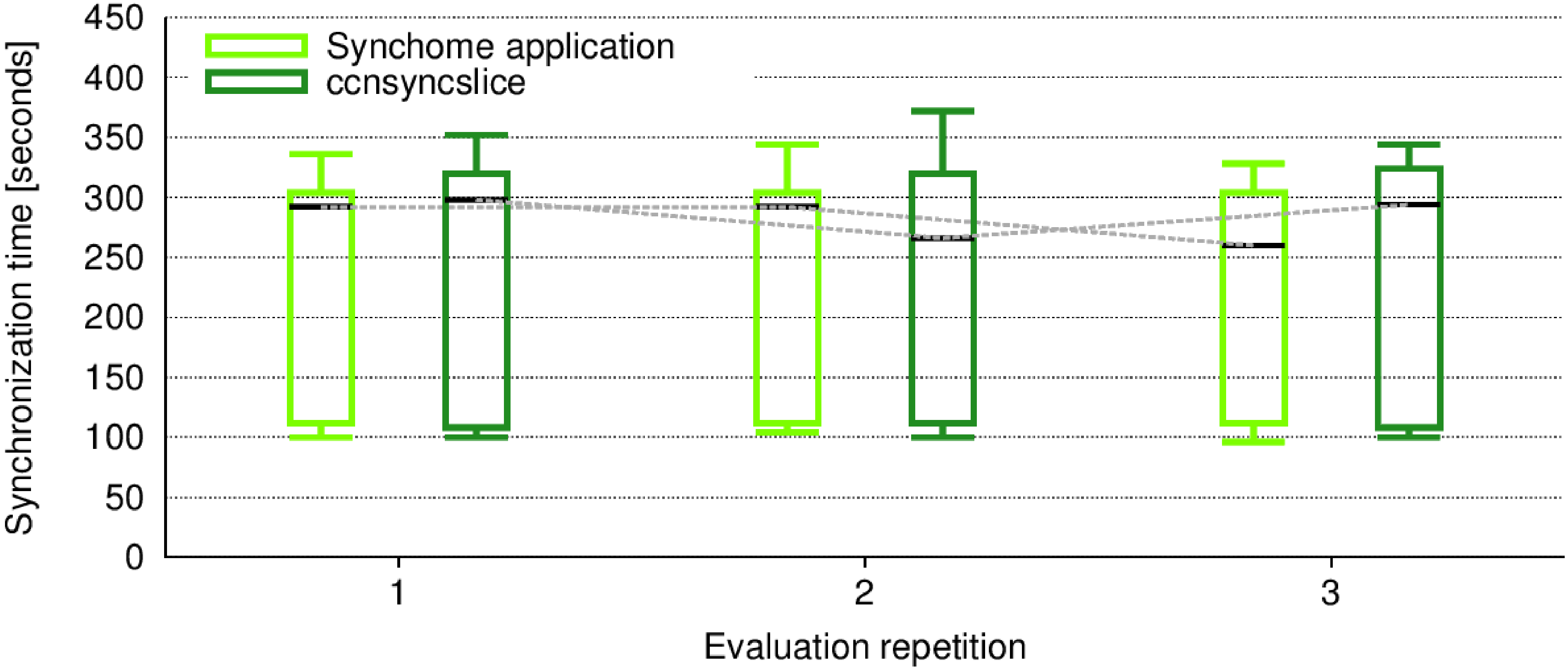
Scenario 1

- Initialize CCNx and repositories
- Add files sized 1, 5 or 10MB to repository
- 5 files each added to 1, 2 or 3 collections
- Run synchome/ccnsyncslice until finished synchronization and stop simulation

2 collections, 1-5-10MB



3 collections, 10MB



Scenario 2

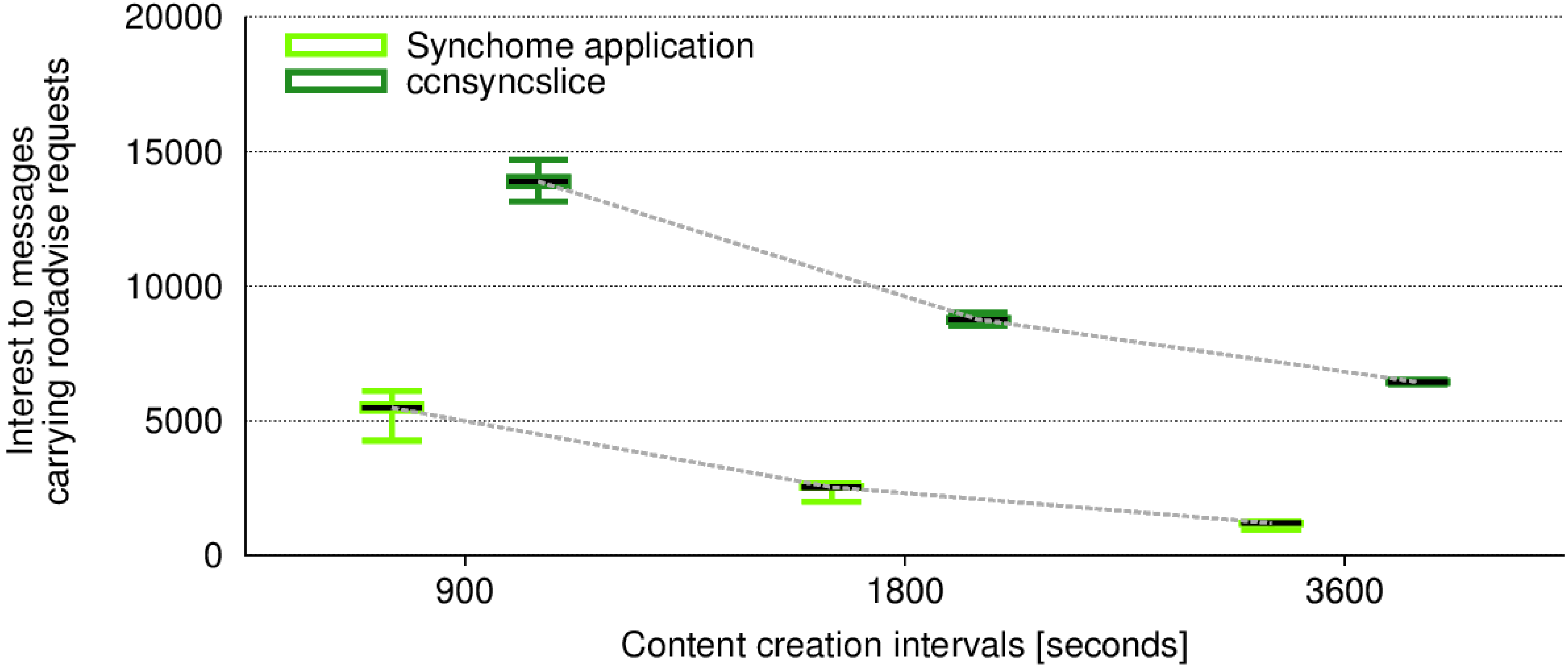
- Synchronize for 24 hours
- Add new content (filesize 1 or 2MB) every 900, 1800 or 3600 seconds
- Ccnsyncslice constantly running
- Synchome is initialized when new content is available and exits when synchronization is finished

Message types

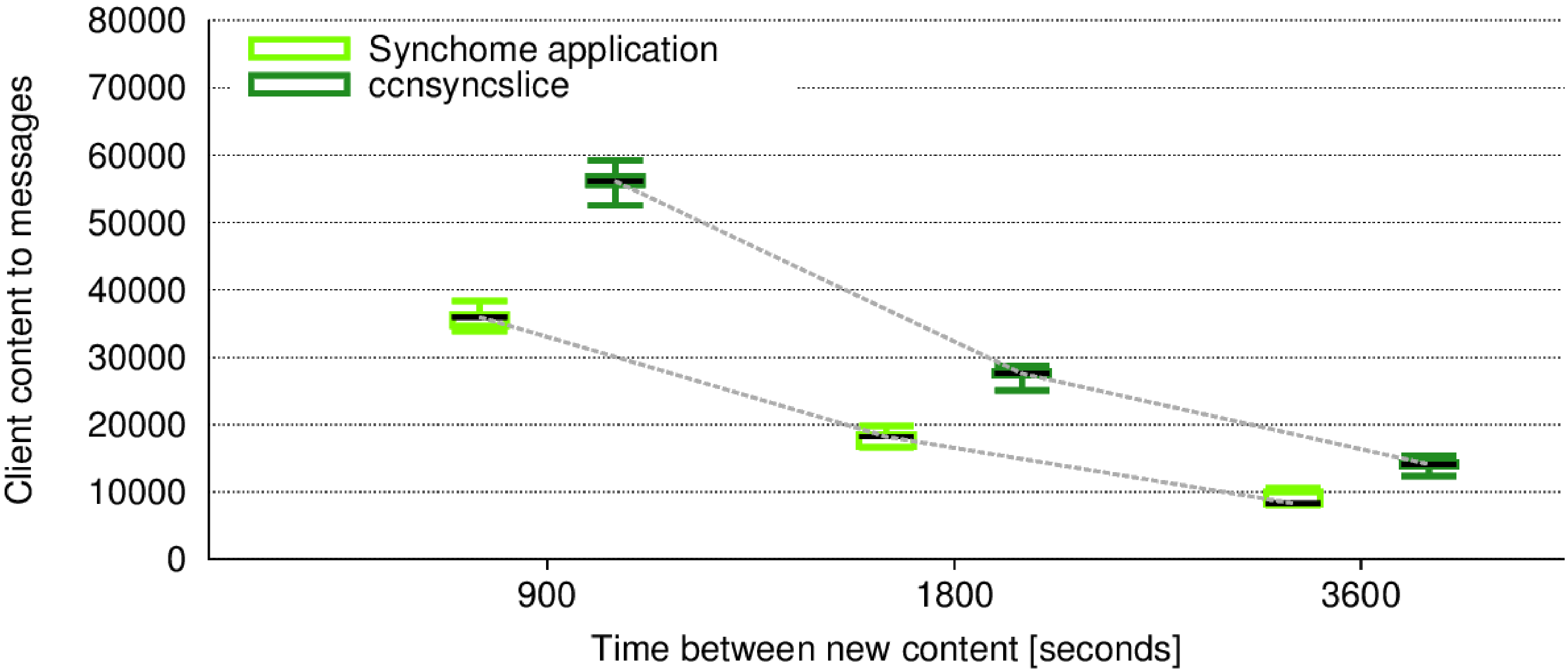
- Interest_to messages: Interests sent out by host. May be Rootadvise or Node fetch Interests, or requests for content.

- Content_to messages: Data messages sent out by host. May carry replies to Rootadvise or Node fetch Interests, or content.

Client Interest messages



Client Data messages



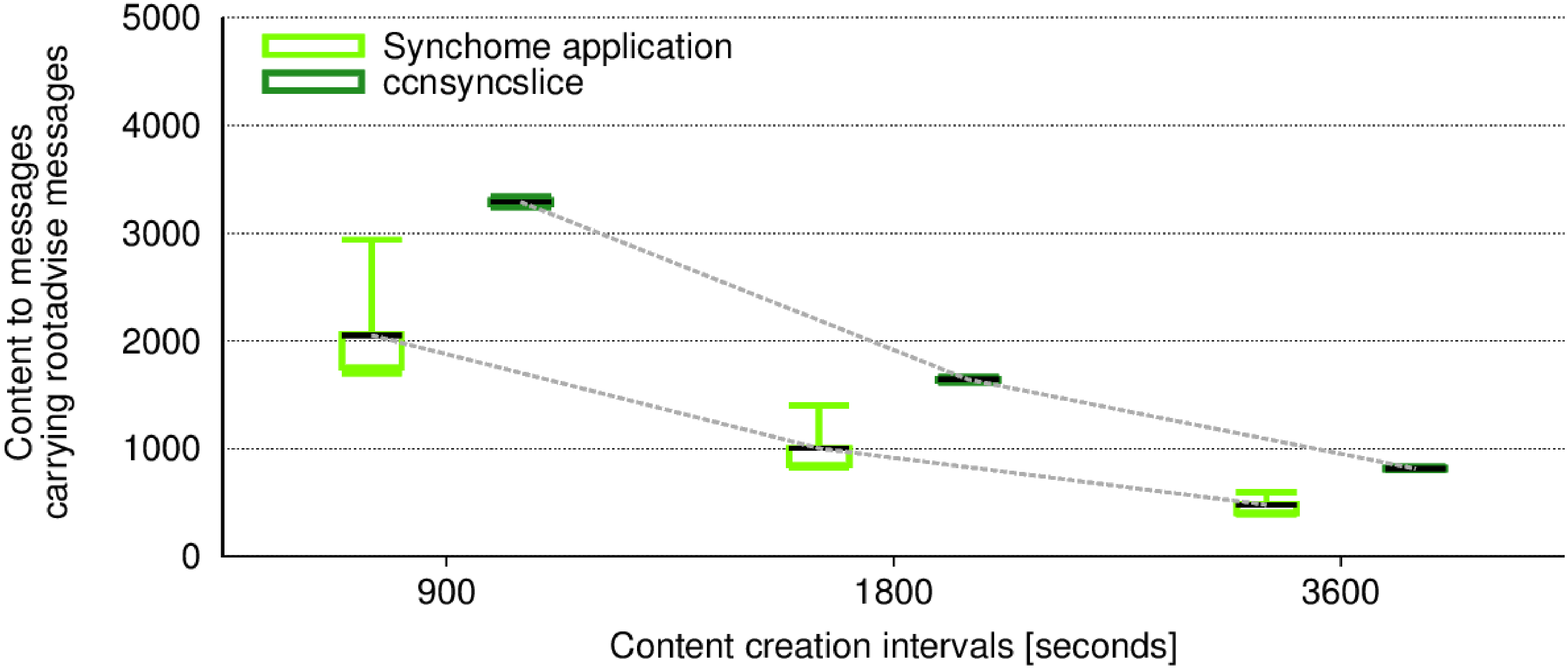
Client Data messages

- Ccnsyncslice sends out around 50% more Data messages than synchome

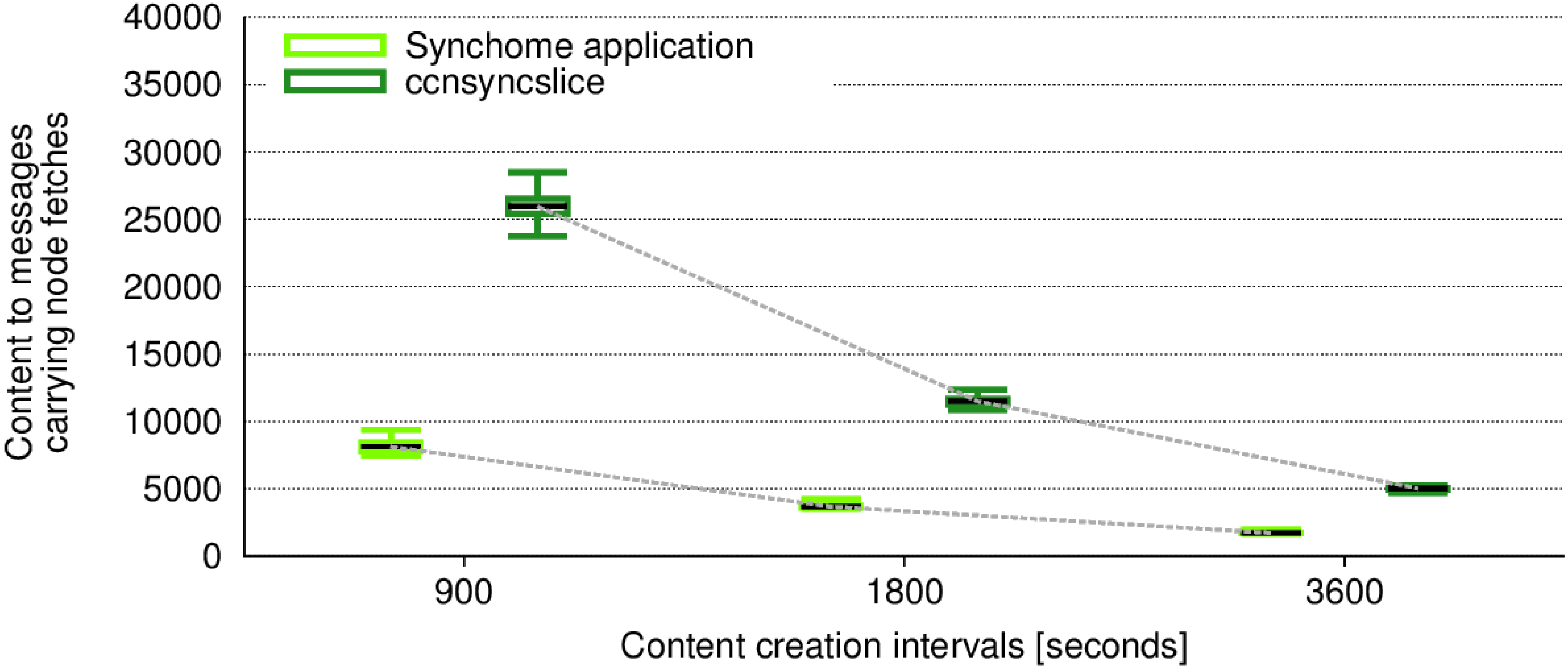
- Number of transferred content segments stays constantly in same range

- So what is causing the increase?

Client Rootadvise replies



Client Node fetch replies



Conclusions

- Synchome generates far less messages than ccnsyncslice.
- The larger the content generation interval, the better synchome performs compared to ccnsyncslice
- Exchange of Node fetch Interests has influence on synchronization time
- Mean runtime difference is nearly negligible between applications. Ccnsyncslice always displays higher maximal runtime.

Questions?