Design of Communication Architecture to Support Stream Data over Content-centric Networking

†Kenya Kawasaki, ‡Shingo Ata, †Masayuki Murata

†Osaka University
Japan

‡Osaka City University
Japan

Outline

- Background
- Motivation and Approach
- Design of Communication Architecture
  - Stream Data Model
  - Content's Name Structure
- Implementation
- Summary and Future Work

Background

- The Internet was designed about 50 years ago
  - Communication is host-based

gap

  - Currently, Internet users are primarily interested in content

- The future Internet is expected to be a post IP networks
  - Content-centric networking (CCN)
  - Communication based on the content name

Motivation

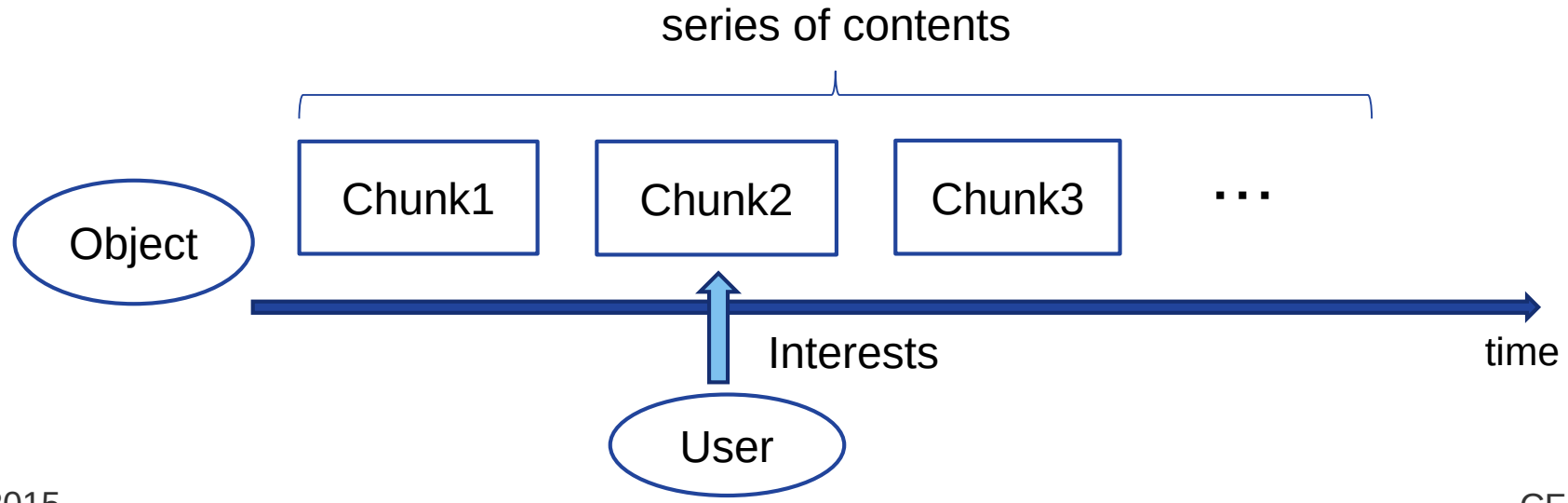
- Various applications over CCN
  - e.g. Media streaming, sensor network, M2M application
  - But most of these studies only implement application-specific protocols
- The complicated architecture is needed to support various application at the same time
- Design a naming architecture that can be used to support various applications in general.

Approach

- Define a communication model called ***stream data***
- Design a naming architecture for supporting stream data delivery
- Implement a prototype of stream data communication

Stream Data Model

- Definition
  - A series of contents with a sequence generated over time by a single source (content provider)
    - e.g. video/audio media data, periodic sensing data
  - Object: device that is data source of stream data
  - Chunk: content that is part of stream data
    - One interest corresponds to one chunk



Name structure for stream data

- Content's name structure
  - <routing prefix>/<identifier>/<control>/<sequence>

example of content's name

data type	content's name
content	ccnx:/osaka-u.ac.jp/loc1/camera/2000-01-01-00-16-32/jpg/QSIF/1/000000027 <routing prefix> <identifier> <control> <sequence>

  - Routing prefix
    - Aggregate routing information
    - Can make routing more efficient
  - Identifier
    - Identify the object

Name Structure for Stream Data

- Content's name structure
  - <routing prefix>/<identifier>/<control>/<sequence>

example of content's name

data type	content's name
content	ccnx:/osaka-u.ac.jp/loc1/camera/2000-01-01-00-16-32/jpg/QSIF/1/000000027 <routing prefix> <identifier> <control> <sequence>

  - Control
    - For controlling the object
      - e.g. Controls the camera resolution, bit rate and sampling rate
  - Sequence
    - Represents the order of chunks
    - Does not have to be evenly incremented number
      - e.g. frame number of a video, timestamp in sensor data

Construction of Content Name

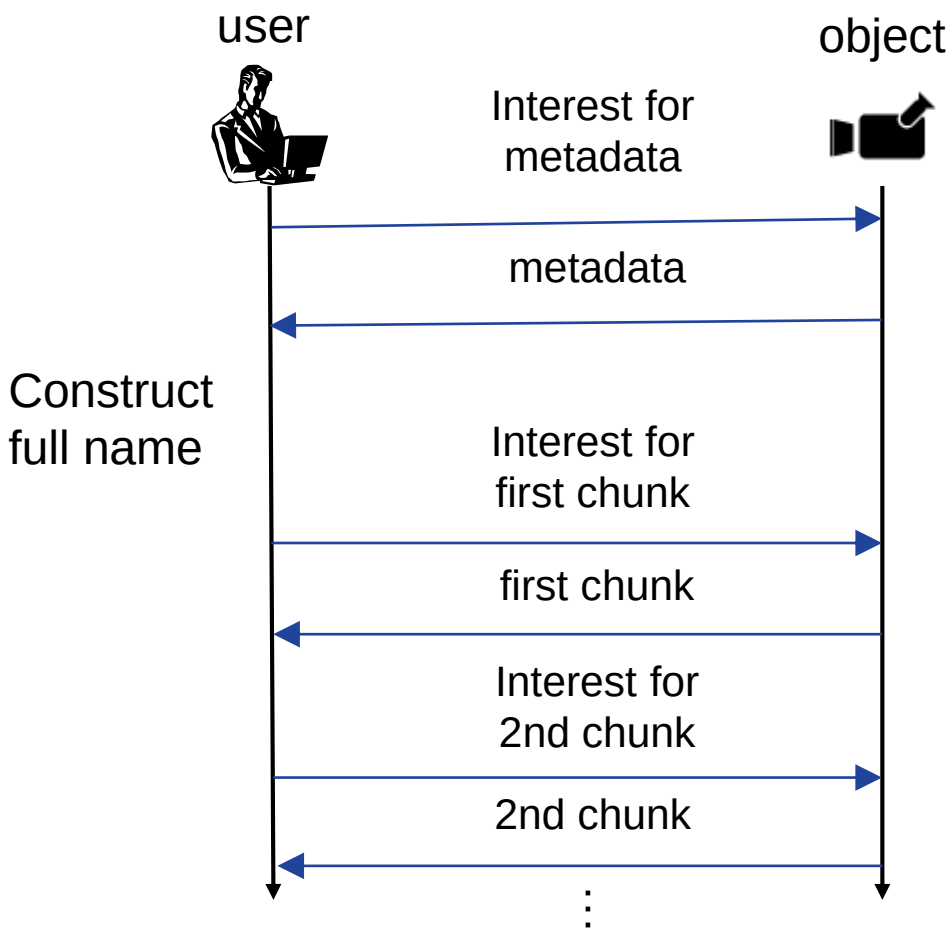
- Available control and sequence are unknown to user initially
  - Use metadata to get control and sequence
- Metadata contains a set of attribute information for the stream data
- Metadata consists of key-value pairs

version: 2000-01-01-00-16-32
codec: jpg
resolution: QSIF | QCIF
:

data type	content's nam	construct
metadata	ccnx:/osaka-u.ac.jp/loc1/camera/metadata	
content	ccnx:/osaka-u.ac.jp/loc1/camera/2000-01-01-00-16-32/jpg/QSIF/1/000000027	
	<routing prefix> <identifier> <control> <sequence>	

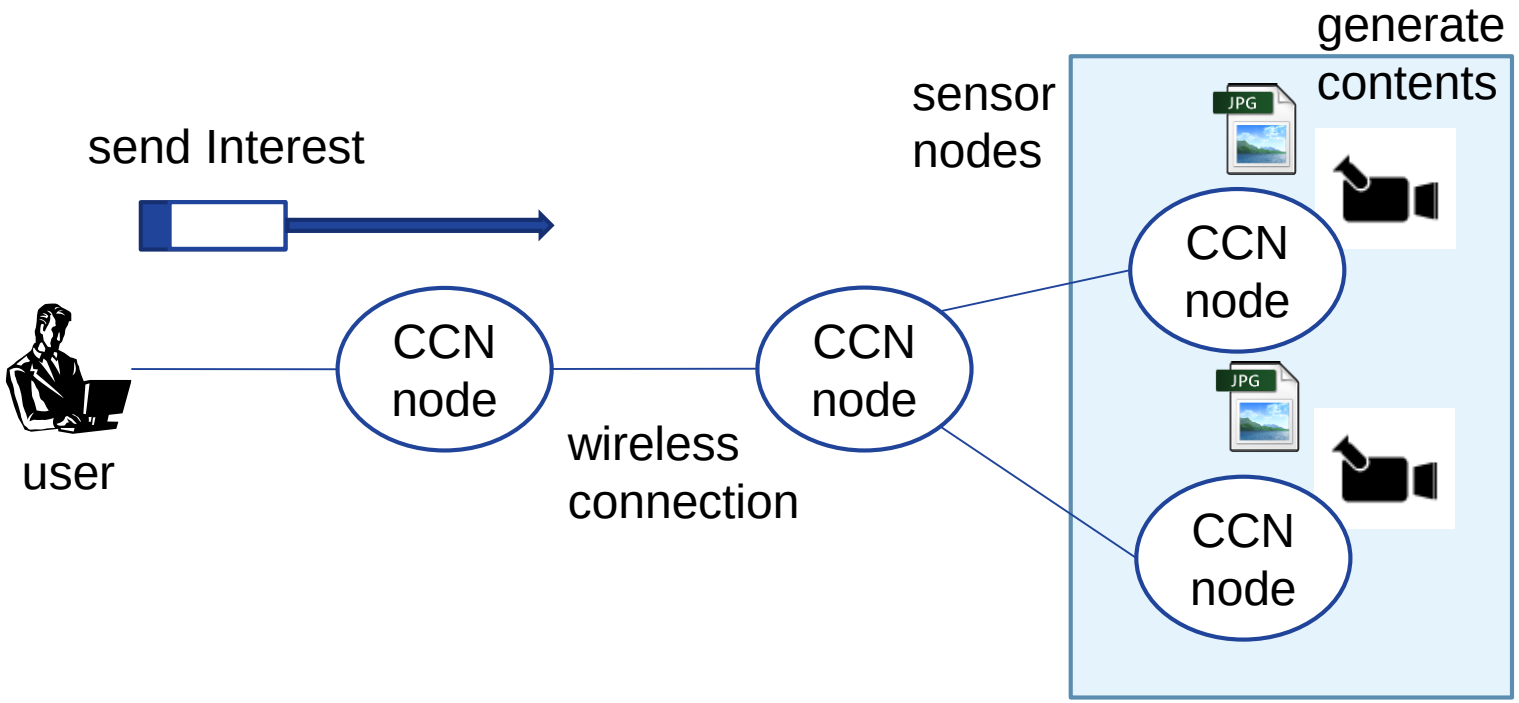
Communication Sequence

- User gets metadata before to start retrieving stream data



Implementation

- We implement stream data delivering system in wireless sensor network
  - Sensor nodes continuously generate contents from image obtained from the attached camera
  - User try to retrieve contents from sensor nodes



Implementation Environment

- Embedded platform “Armadillo-420” is used for CCN nodes
  - Attach USB Web cameras to sensor nodes
- We use CCNx protocol suite[6] to implement CCN nodes

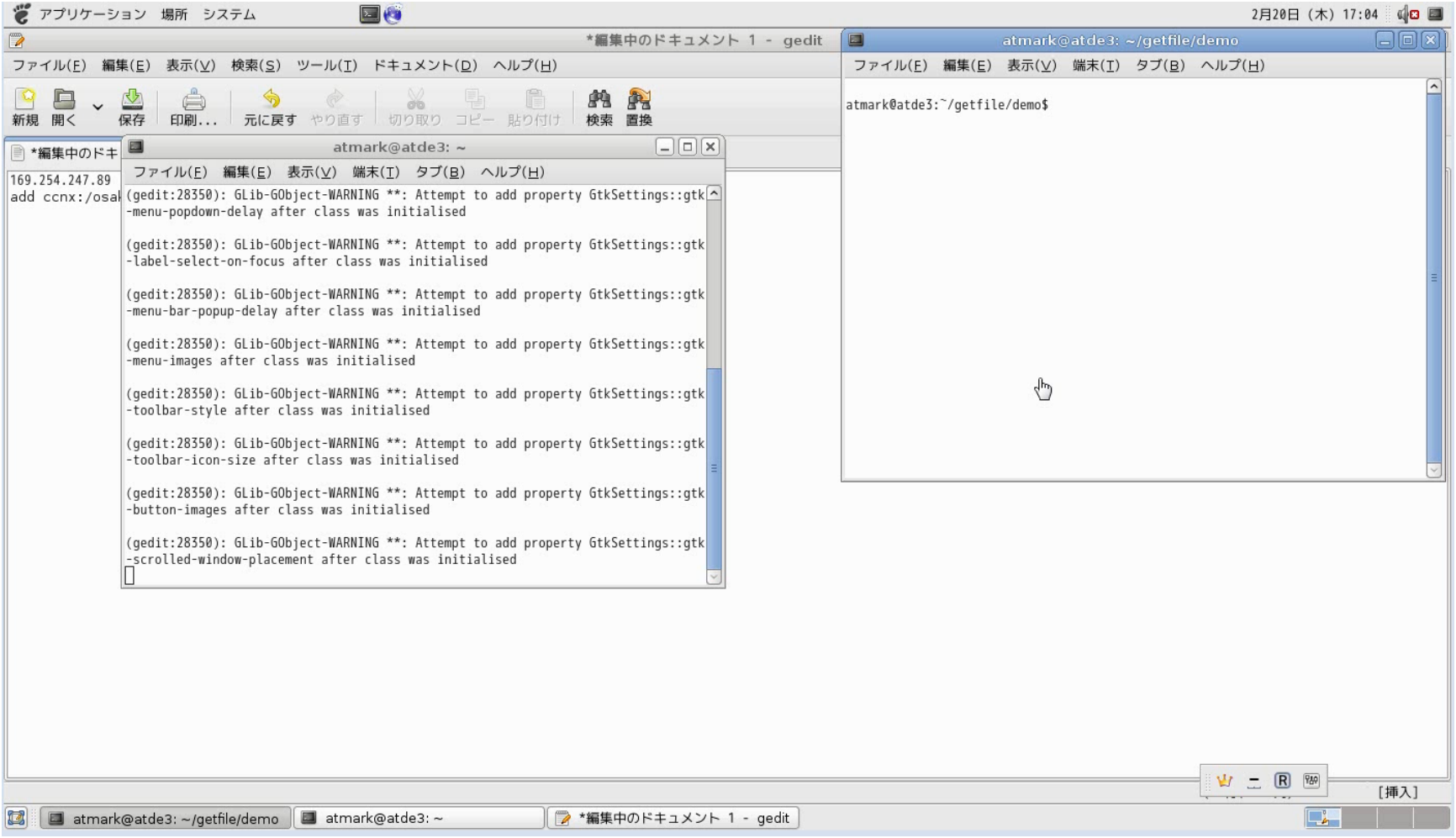
CCN node and sensor (camera)

Hardware spec for Armadillo-420

Processor	Freescale i.MX257
CPU core	ARM926EJ-S
CPU core clock	400 [MHz]
Bus clock	133 [MHz]
RAM	64 [MB]
Flash memory	16 [MB]
Wireless LAN	IEEE 802.11b/g/n
USB	USB 2.0

[6] “CCNx,” PARC, 2013, http://www.ccnx.org

Demonstration



Summary and Future Work

- Summary
  - We designed the communication architecture to support stream data
    - Define the stream data model
    - Design the content's naming structure
  - We implemented a prototype of stream data delivering system

- Future work
  - Evaluate the designed system
  - Consider more detailed M2M data model