CCN Traffic Optimization for IoT

Jérôme François, Thibault Cholez, Thomas Engel

UNIVERSITÉ DU LUXEMBOURG

SnT

SNT

Outline

1. Introduction
2. CCN Background
3. Problem Definition
4. Solution
5. Evaluation
6. Conclusion

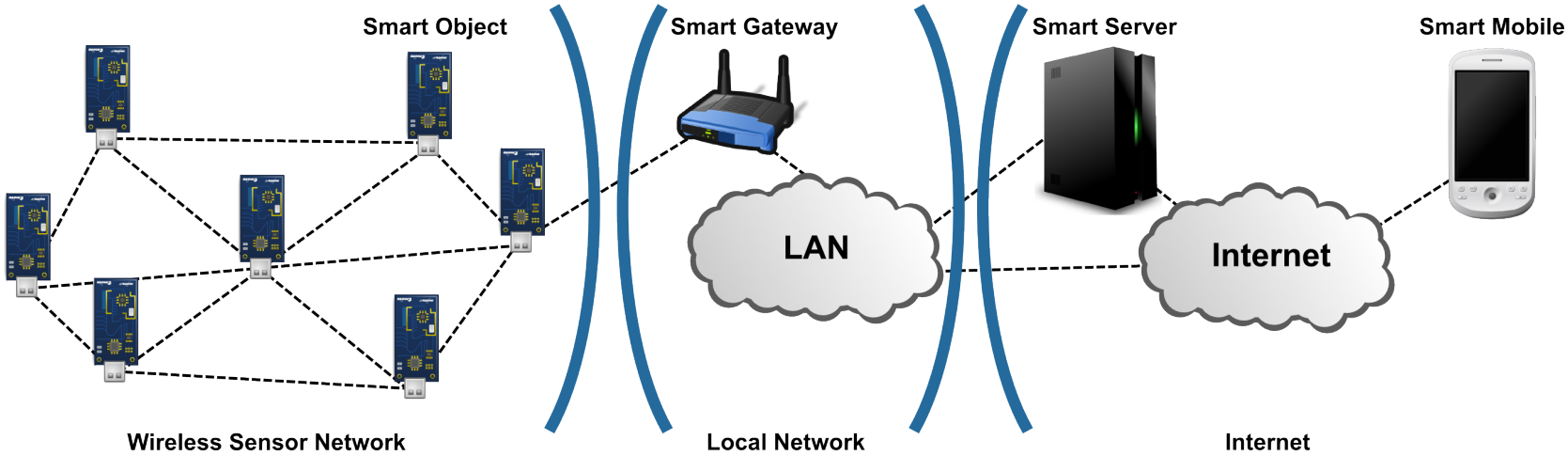
SNT

Outline

1. Introduction
2. CCN Background
3. Problem Definition
4. Solution
5. Evaluation
6. Conclusion

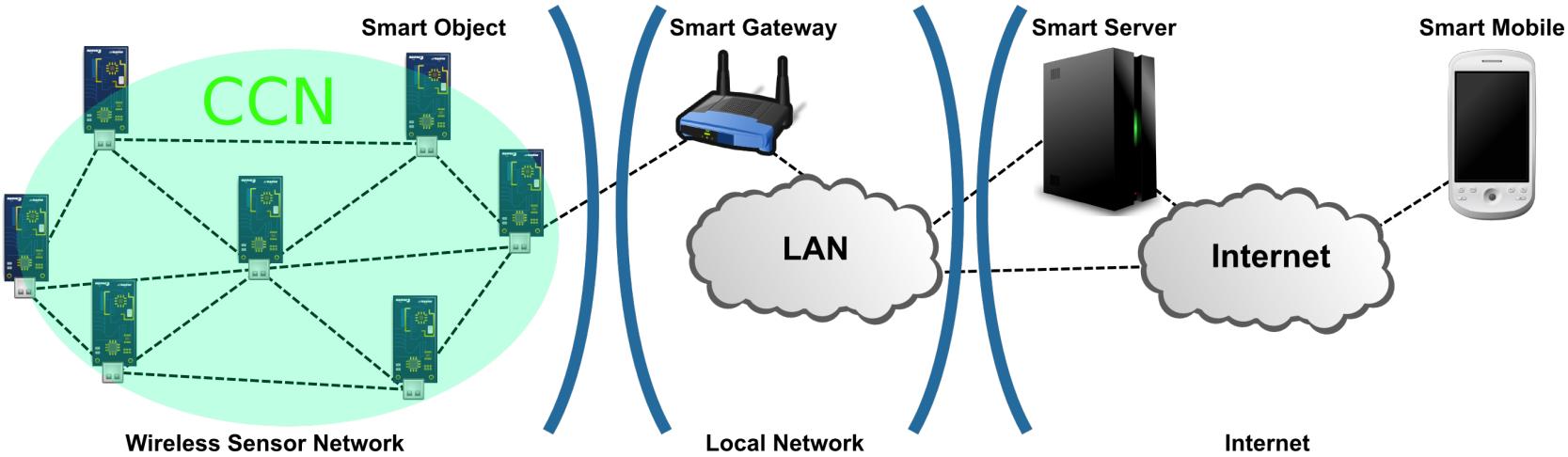
Context

- Internet of Things
  - multi sites network with sensors and gateways
  - shared infrastructure of multiple owners for multiple service providers
  - reliability, security, privacy... requirements



Context

- Internet of Things
  - multi sites network with sensors and gateways
  - shared infrastructure of multiple owners for multiple service providers
  - reliability, security, privacy... requirements



SNT

CCN for Sensors

- Why?
  - CCN designed for distributed networking → scalability
  - ICN/CCN provides already many required basic functionalities (security, naming)
  - → alleviates the need of additional protocol, *e.g.* resource discovery protocols
  - Contiki implementation available
- but:
  - CCN: pull-based mode (on demand information retrieving)
  - many IoT use cases where: push based model for sensor (regular sensing or event based)
- → can we still use CCN for sensor communications on a push based manner?

SNT

Outline

1 Introduction

2 CCN Background

3 Problem Definition

4 Solution

5 Evaluation

6 Conclusion

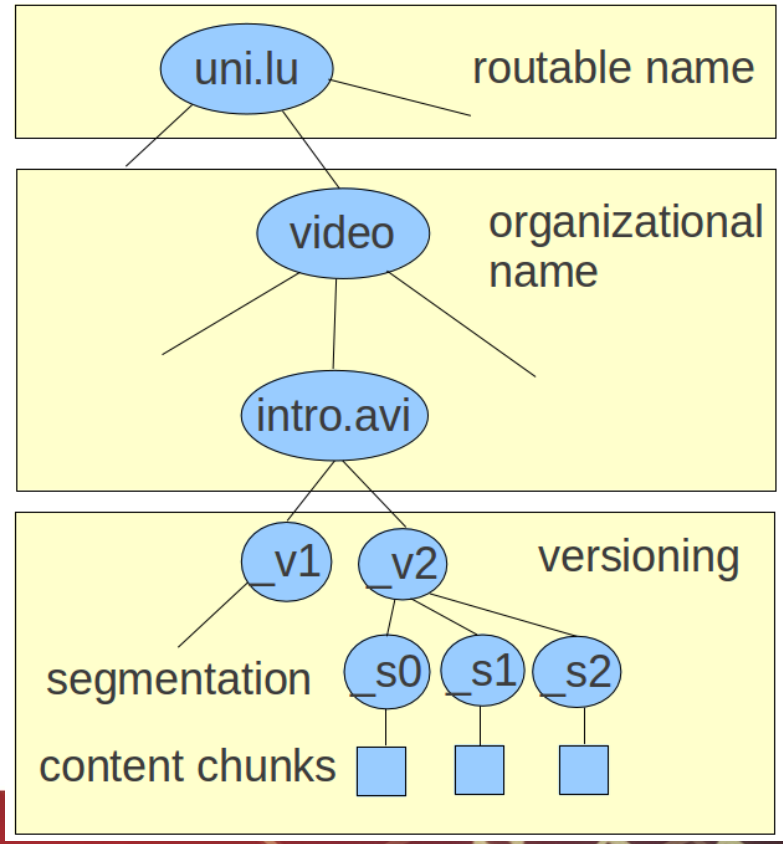
SNT

Basics

- Key ideas
  - Data as a name, not a location
  - Data is directly requested at the network level (no more DNS: more security; less delay)
  - Anybody with the data can answer
  - Data is authenticated, not the connections it traverses
  - + caching at intermediate routers
- 2 types of packets *Interest* and *Data*
  - Consumer driven: broadcast Interest, waits for Data
  - Data transmitted in response, consumes the Interest

Hierarchical naming

- Routable prefix, persistence of the names
- Hierarchical aggregation for fast routing and forwarding
- Dynamic generation of content
- Context-aware resolution (*/ThisRoom/projector*)



Message sequence

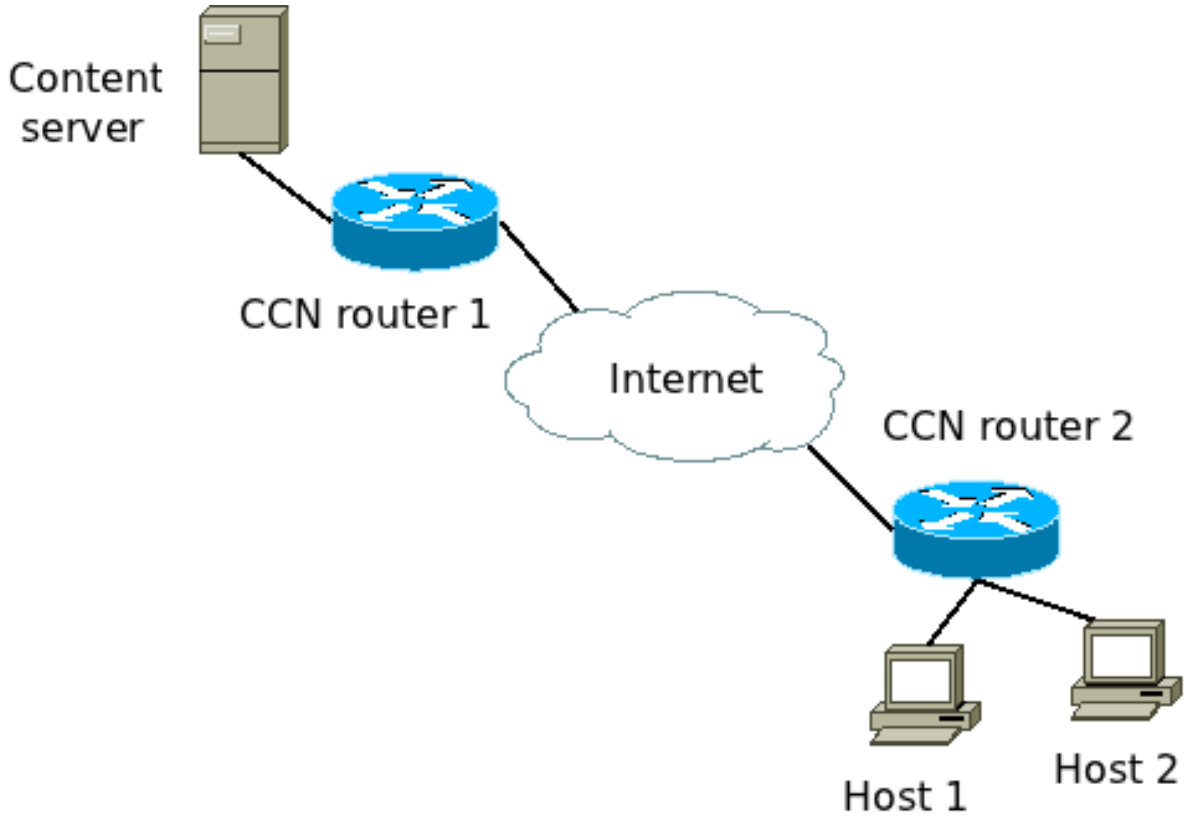


Figure: Scheme of a minimal Content-Centric network

Message sequence

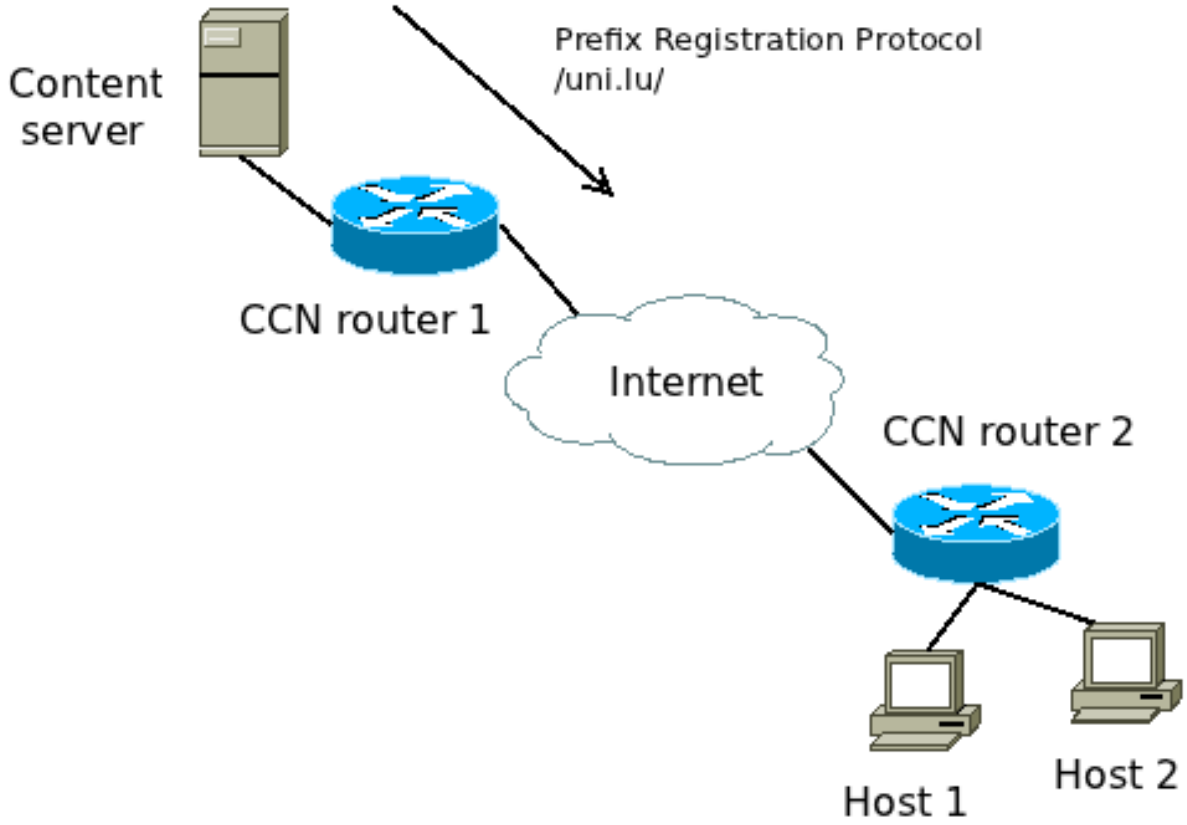


Figure: Scheme of a minimal Content-Centric network

Message sequence

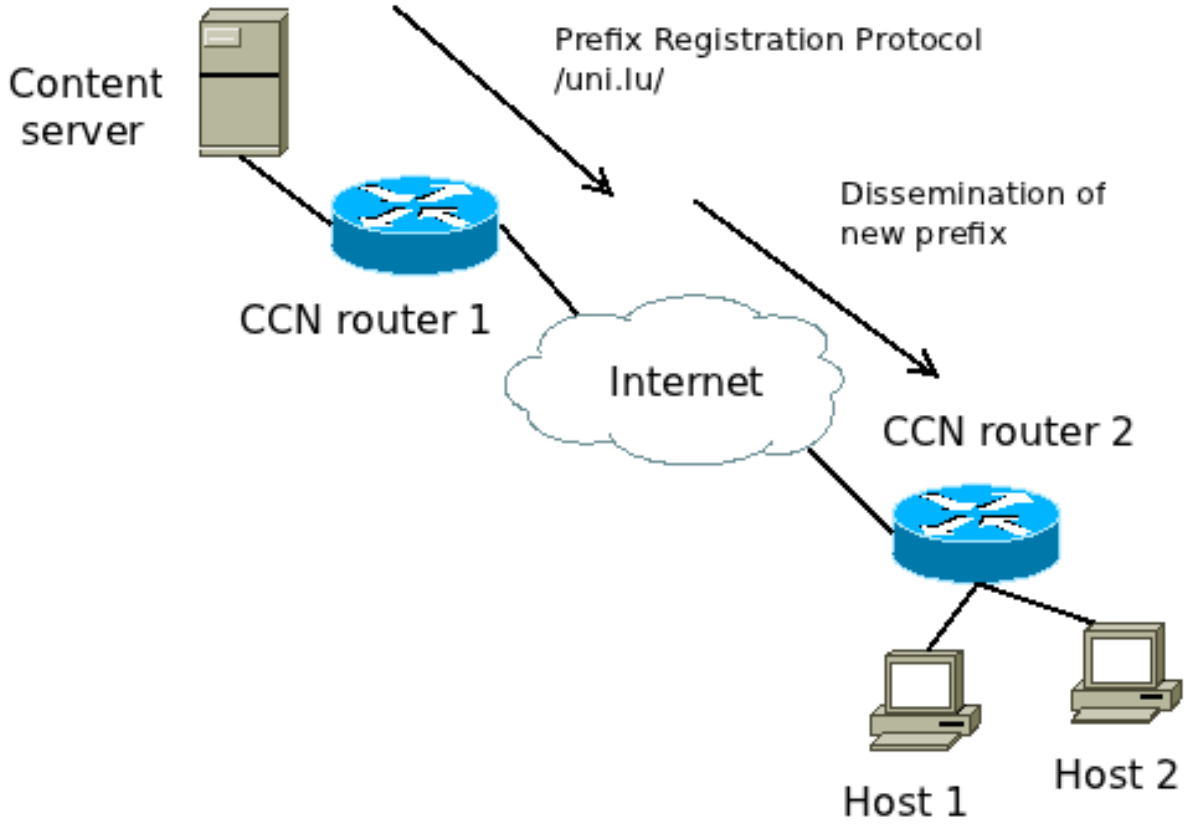


Figure: Scheme of a minimal Content-Centric network

Message sequence

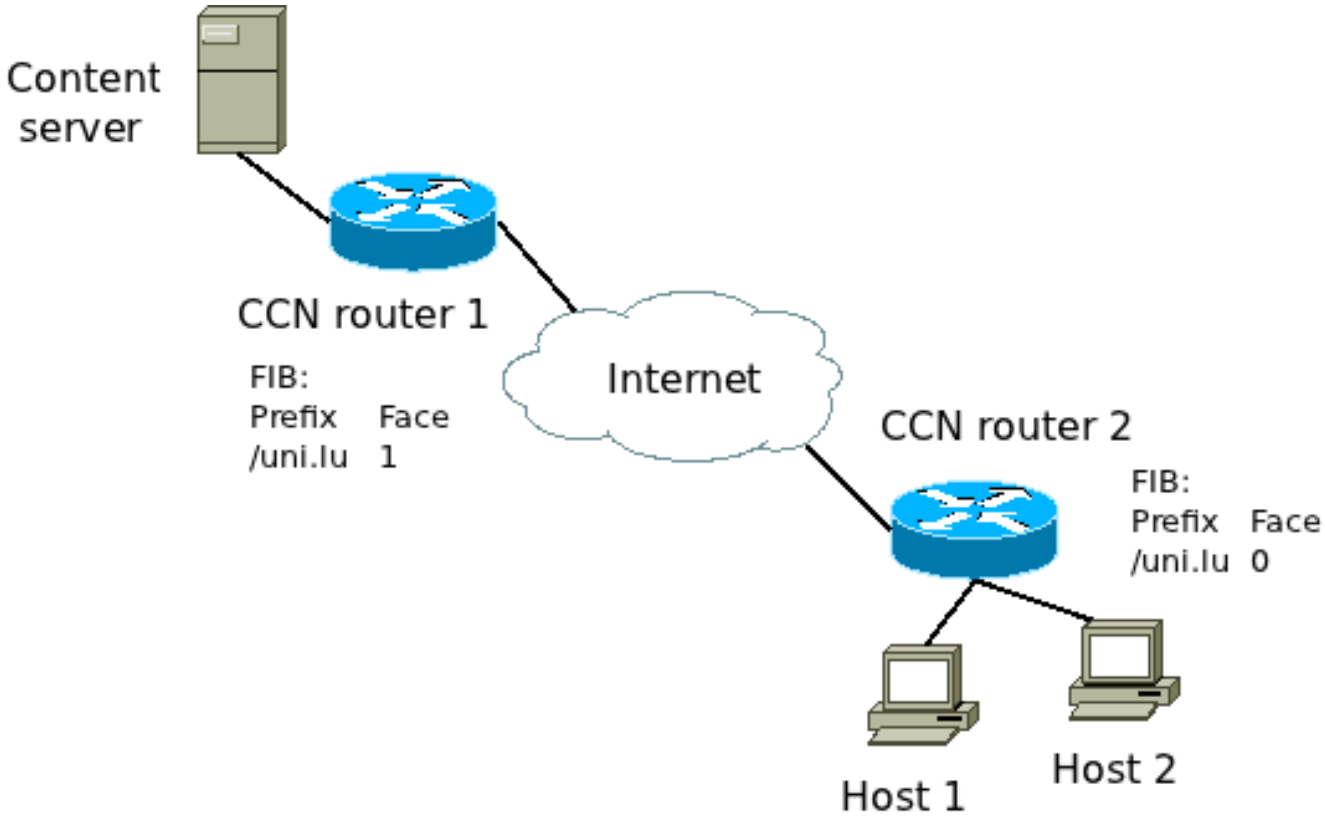


Figure: Scheme of a minimal Content-Centric network

Message sequence

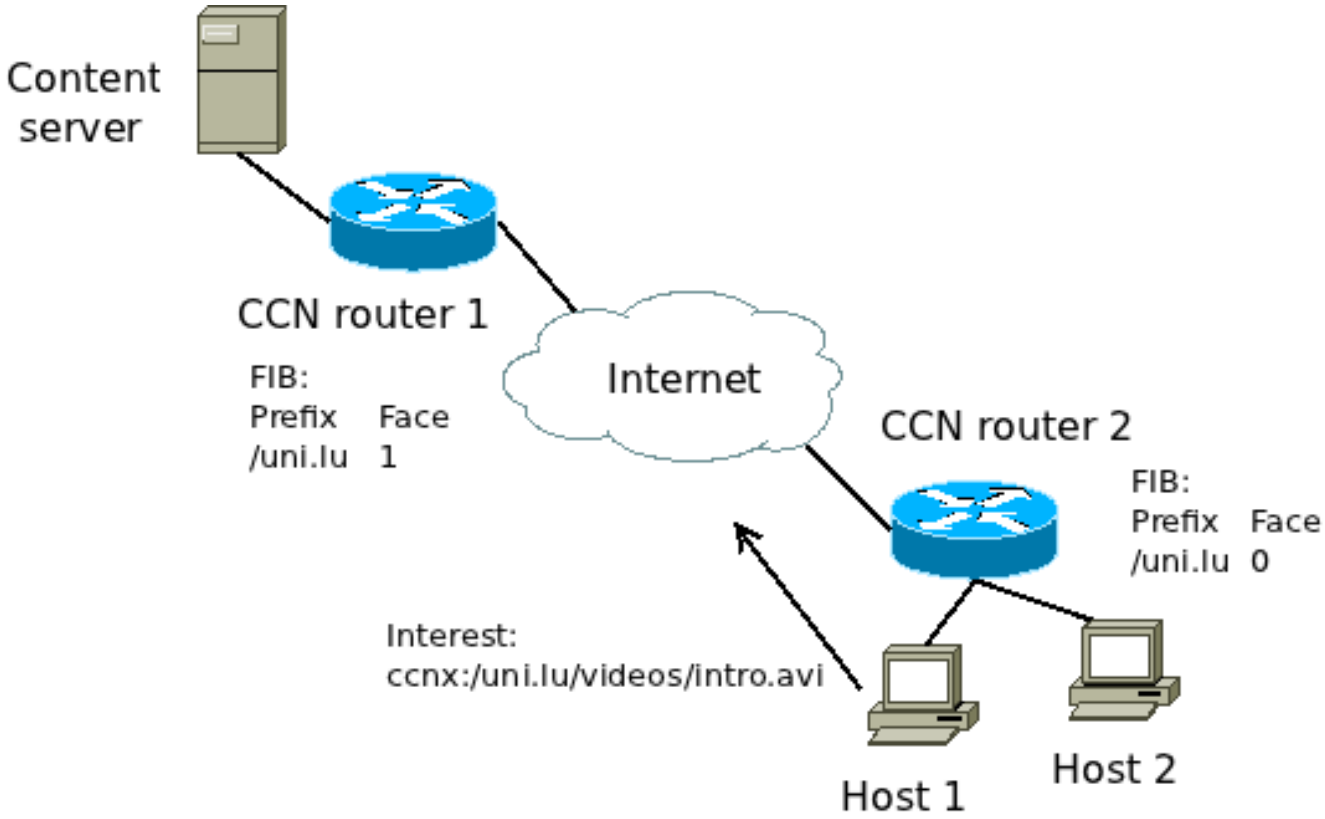


Figure: Scheme of a minimal Content-Centric network

Message sequence

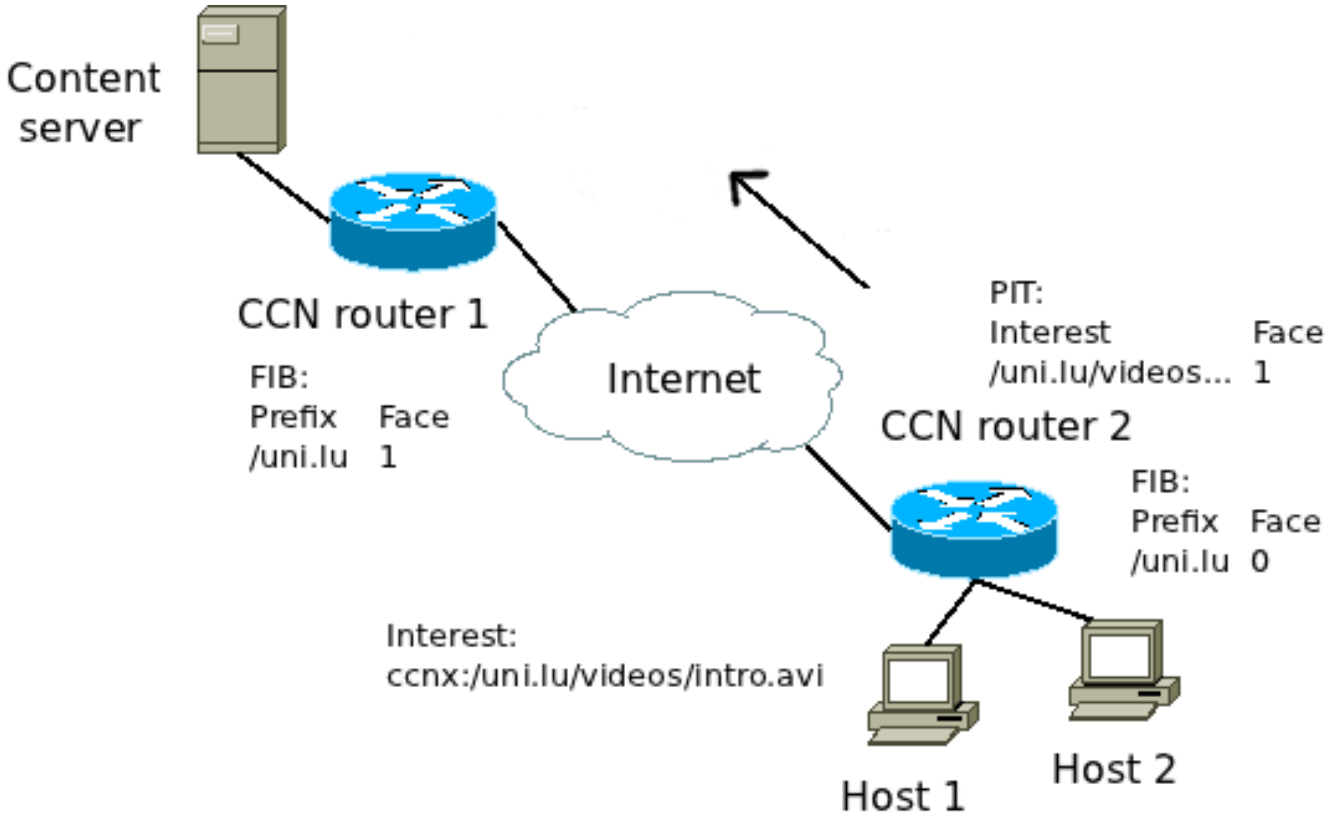


Figure: Scheme of a minimal Content-Centric network

Message sequence

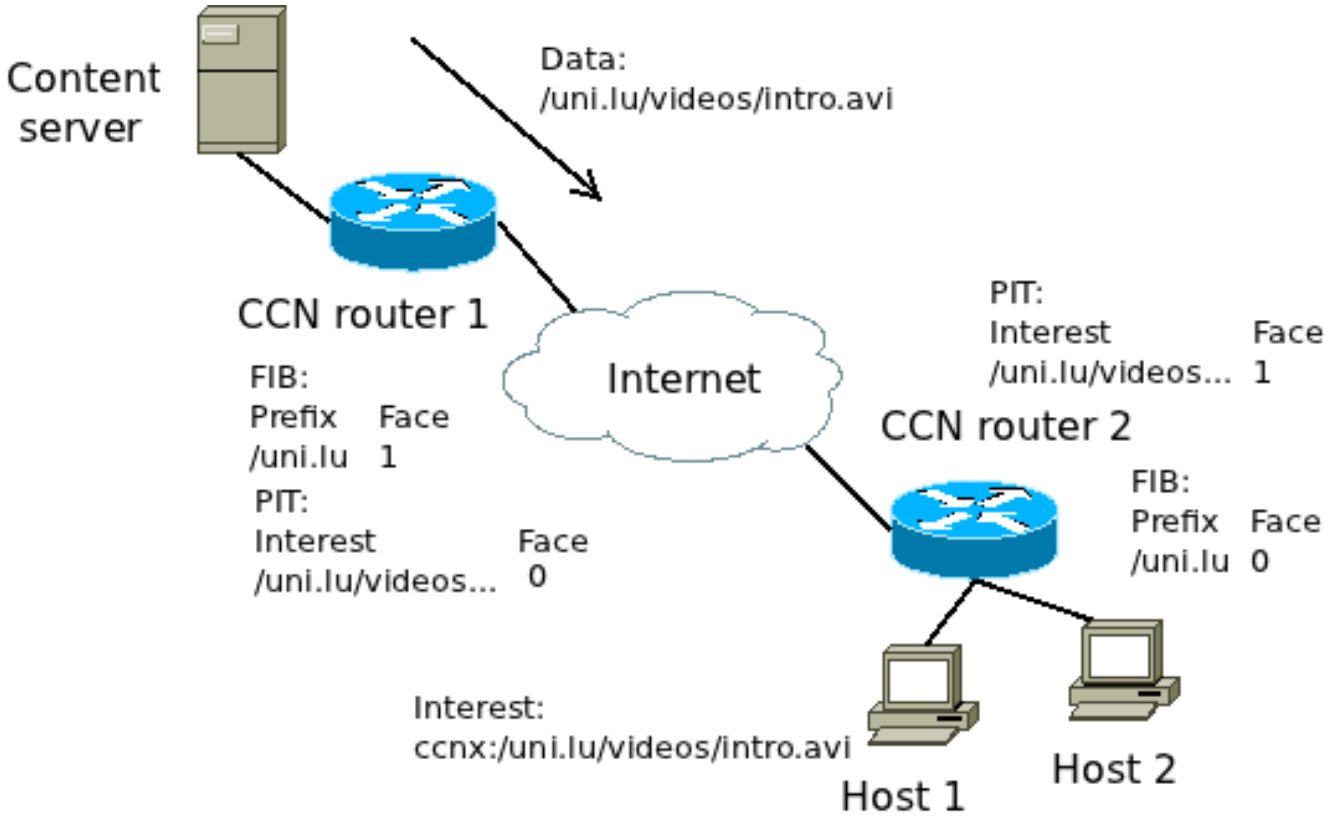


Figure: Scheme of a minimal Content-Centric network

Message sequence

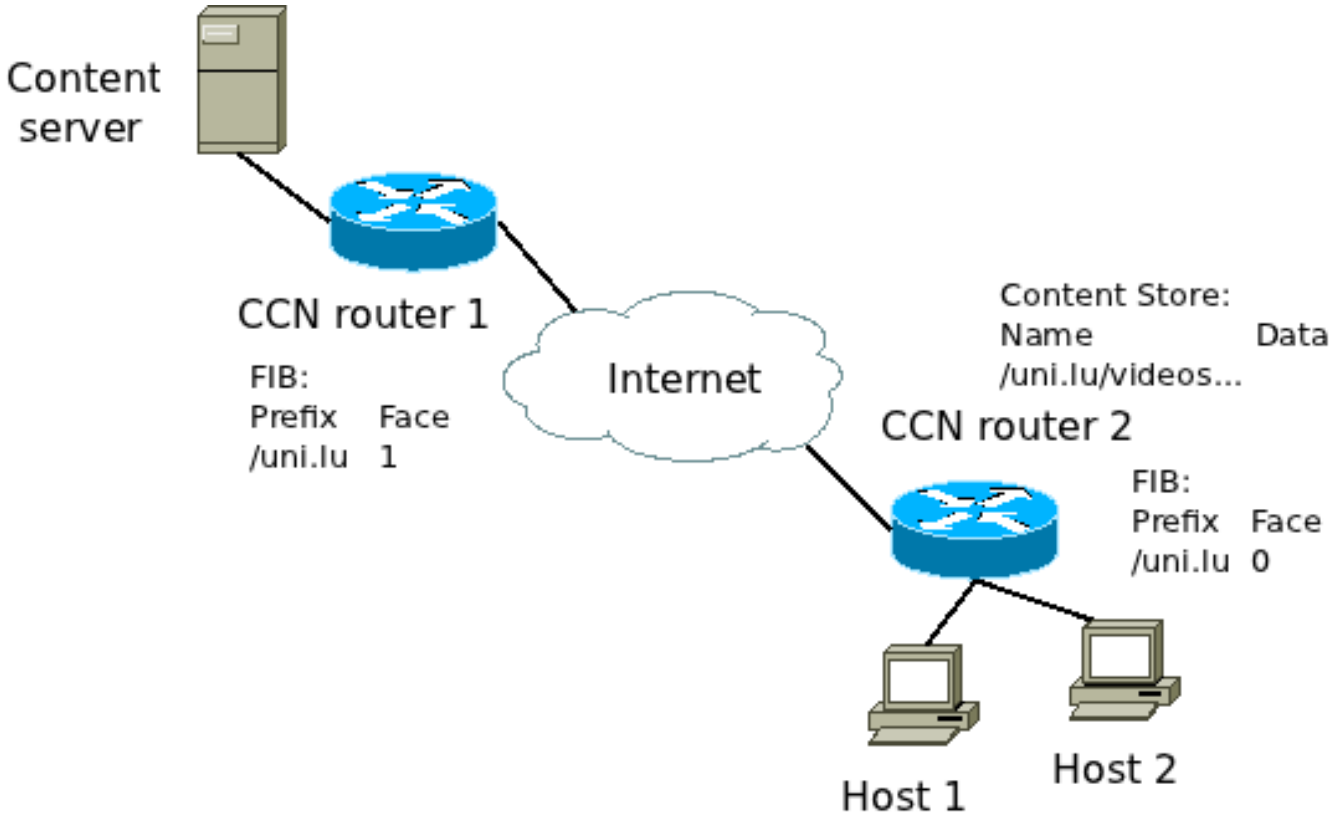


Figure: Scheme of a minimal Content-Centric network

Message sequence

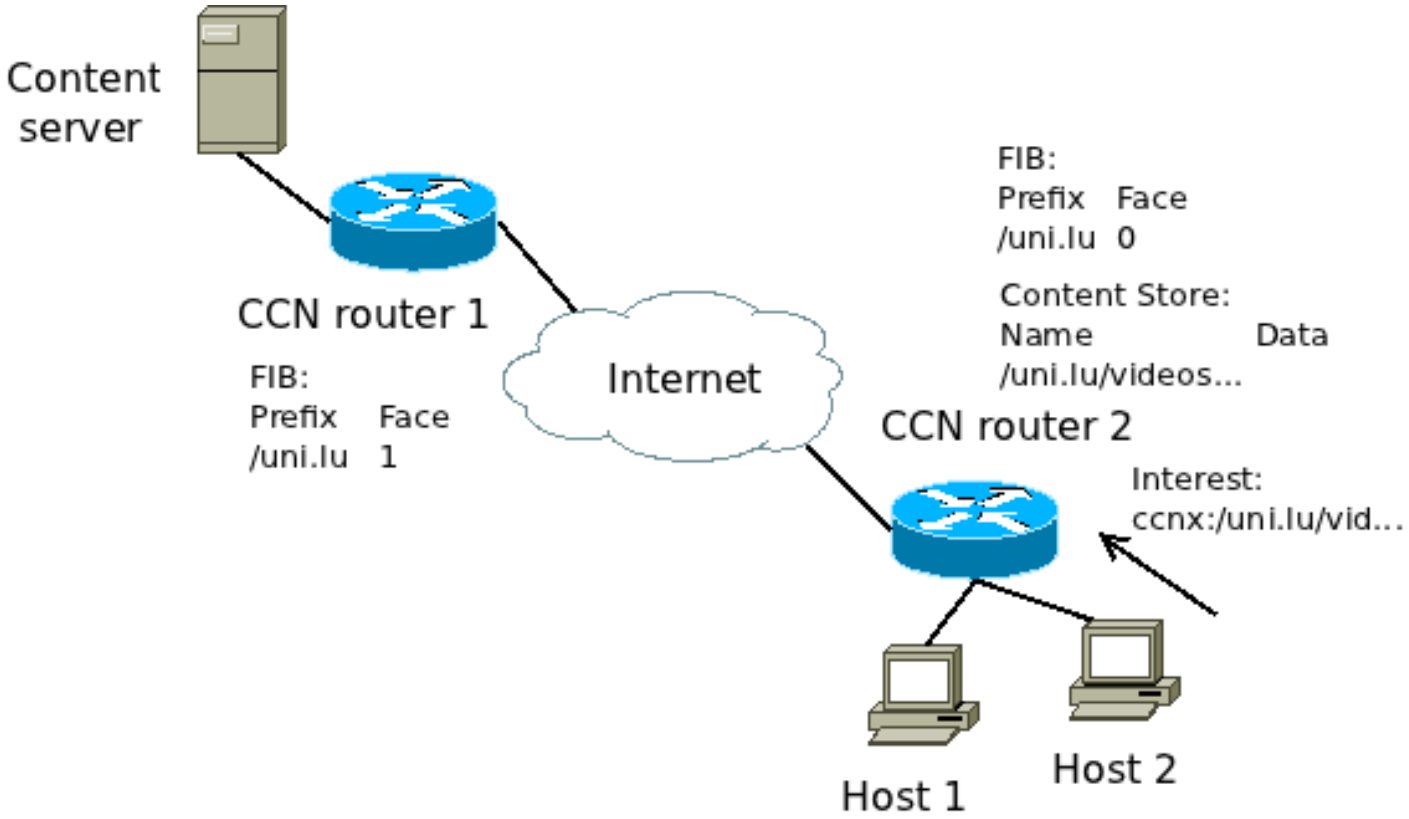


Figure: Scheme of a minimal Content-Centric network

Message sequence

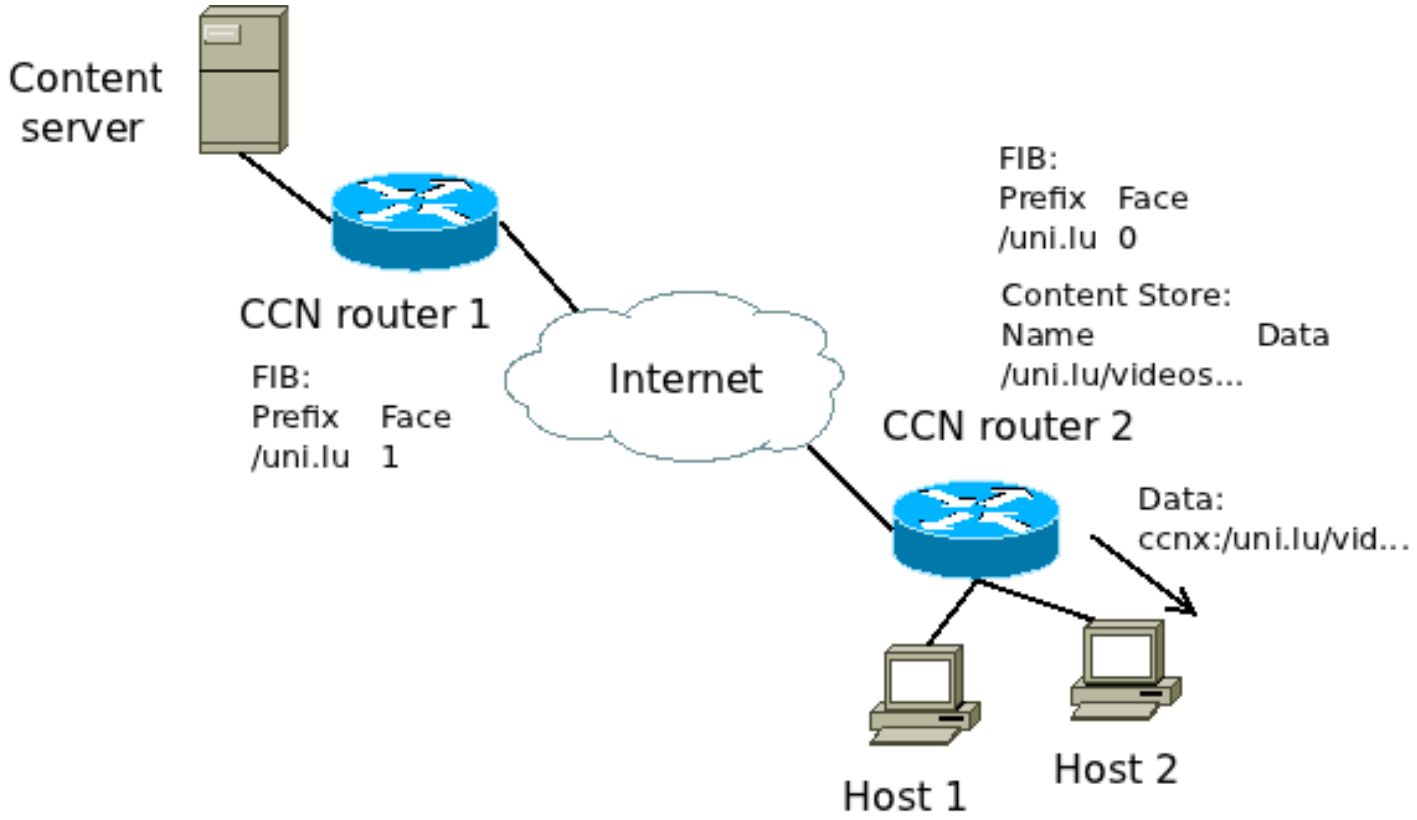


Figure: Scheme of a minimal Content-Centric network

SNT

Outline

1 Introduction

2 CCN Background

3 Problem Definition

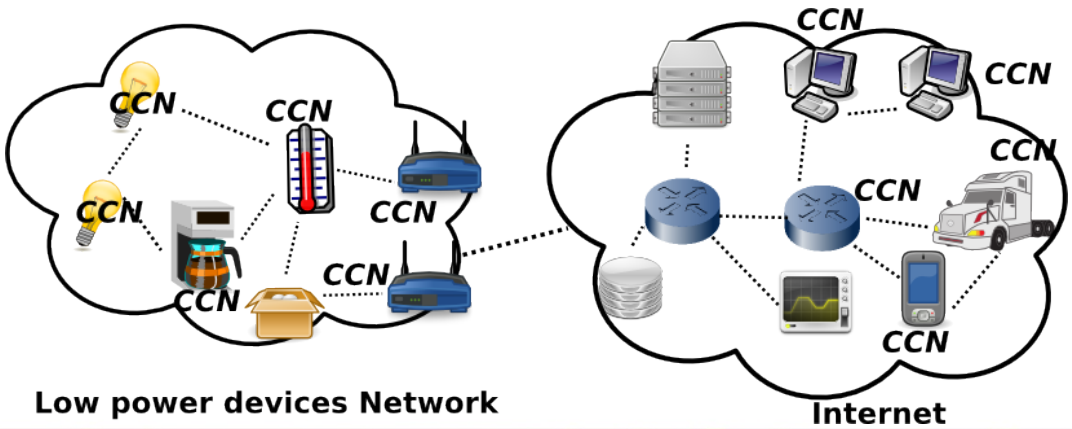
4 Solution

5 Evaluation

6 Conclusion

CCN deployment

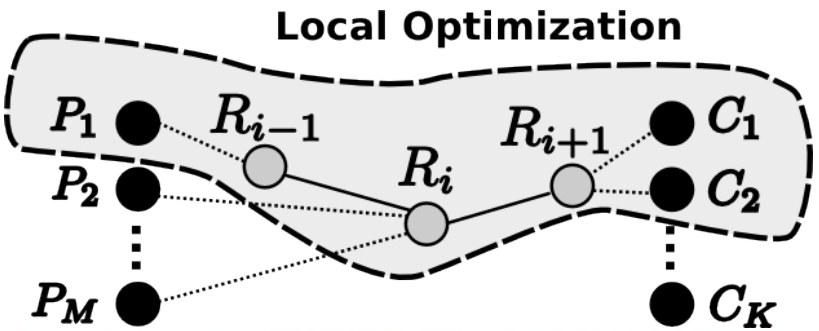
- CCN faces → transparent IP network transversal
- CCN at all sides: sensors, servers, user devices
- context of this work = sensor communication optimization (low power devices)
- → retrieved sensed information
  - push based communication
  - regular updates (every 10 sec, 5 minutes)
  - multiple consumers (with different update frequencies)



SNT

Formal problem

- Limited resources (CPU, memory, energy...)
  - no broadcast / no multicast to every consumer at every time unit
  - example: temperature update every 10 sec, service need update every 5 minutes → overhead = 29 messages
- Problem definition
  - M producers, K destinations, CCN routers
  - Disjoint sensed information → local problem
    - 1 producer, N consumers $C = c_1, \ldots, c_N$ with sampling periods i_j
    - R_i: Cap_i (available ressources)
    - objective: $min(msg(R_i))$



SNT

Outline

1 Introduction

2 CCN Background

3 Problem Definition

4 Solution

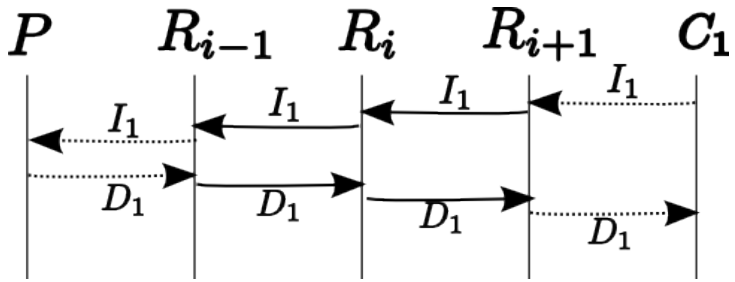
5 Evaluation

6 Conclusion

SNT

Push-based communication

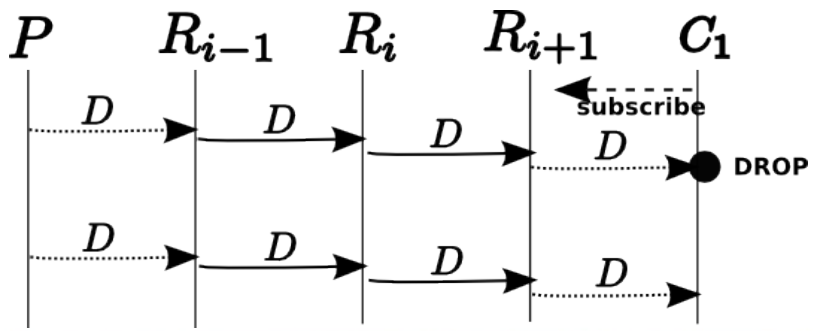
- CCN data forwarded when an interest is emitted = pull model



- SotA propositions
  - Jacobson *et. al*, ACM ReArch 2009: creation of a rendez-vous point → transaction model, individual demand from the producer (no subscription mechanism)
  - Carzaniga *et. al*, ACM SIGCOMM ICN 2011: dissemination of ephemeral messages → a new unicast table (no multicast) based on IP routing for avoiding routing loops

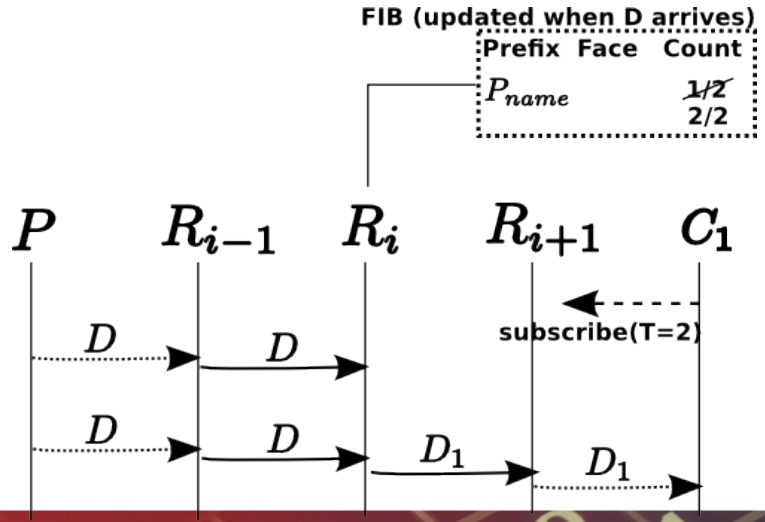
Push-based communication

- Approach overview
  - subscription = content publications
  - sensor value disseminations
    - → use the standard FIB to disseminate sensor value
    - interest message towards the consumer content name prefix + value as its suffix
    - */roomA/thermometer/value/20*
    - → but does not work as there is no content in the other way
    - use FIB for forwarding not requested content → no PIT update
    - → routing loop (interests are not consumed anymore → *last_seen* flag in FIB



Simple filtering

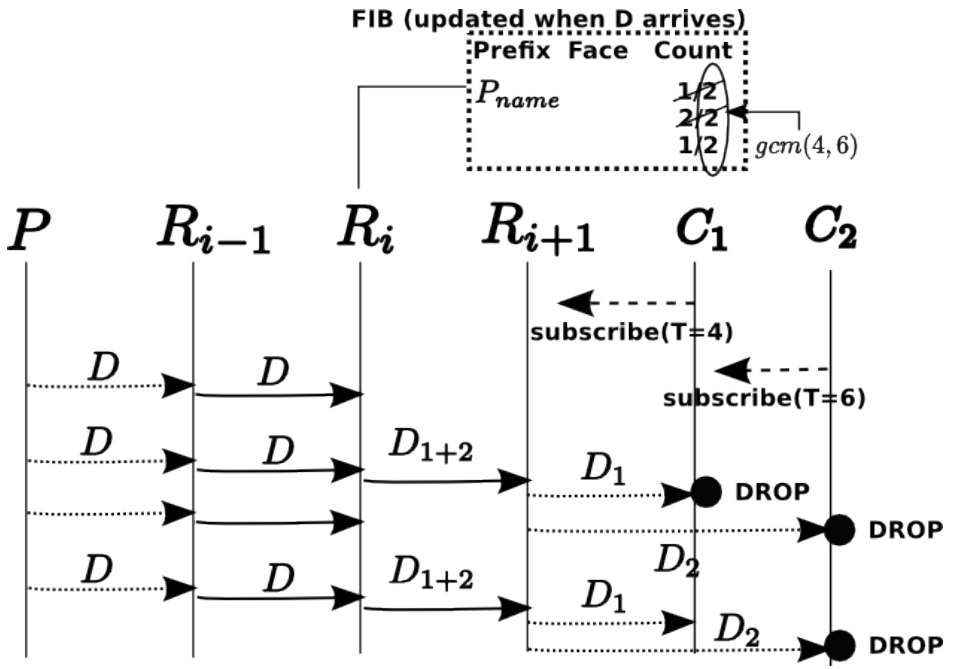
- Avoid sending useless dropped messages
  - simple filtering
    - the router only forwards data to C_i at the period t only if the consumer C_i has expressed its interest regarding this period
    - $\rightarrow$ t is a multiple of the sampling period i_i
    - $\rightarrow$each C_i has to specificity its sampling period to R_i (smart router)
    - need a counter in the FIB, one counter per subscriber = optimal strategy



Smart forwarding – gcd

- Use a limited number of counters Cap_i
- Extreme case $Cap_i = 1$
  - apply partially simple filtering
  - some packets still dropped $\rightarrow$ minimize it

- remove identical subscribed periods i_i
- if multiples in subscribed periods and keep the smallest, *e.g.* $i_1 = 2$ and $i_2 = 4 \rightarrow$ forward packets every 2
- extend the previous concept using the GCD (Greatest Common Divisor)



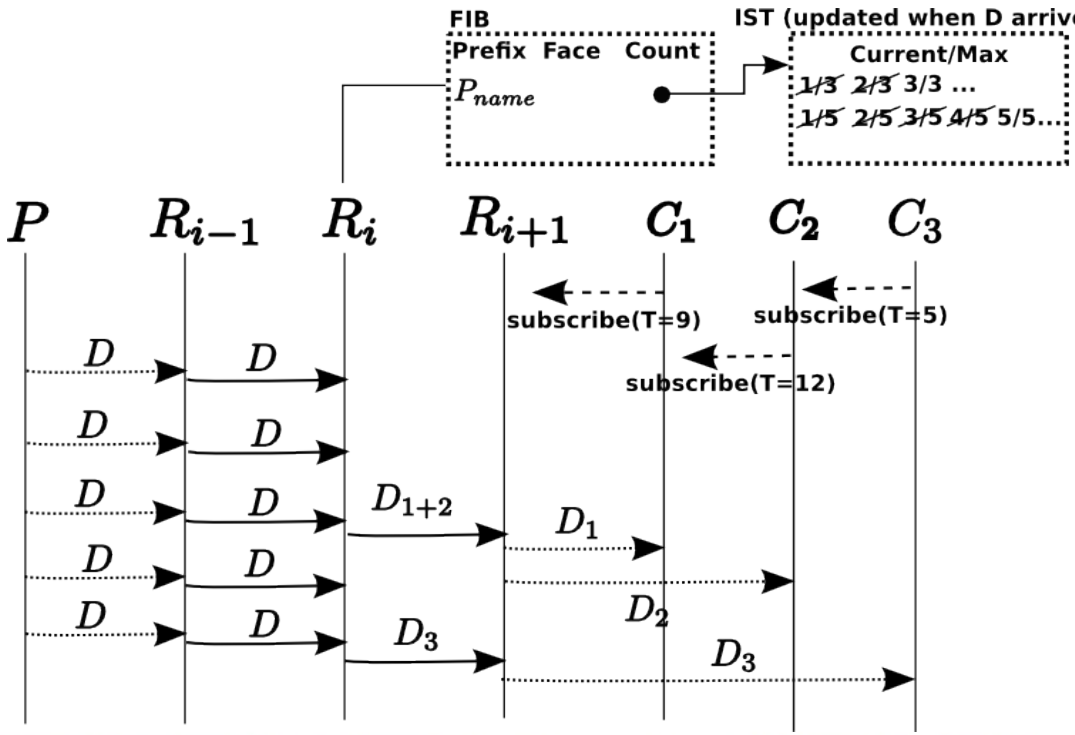
Smart forwarding

- More counters, $Cap_i > 1$
  - IST (Information Sampling Table) for tracking multiple counters
  - guarantee that each C_i receives the packets as subscribed or more
  - $\rightarrow$ select $G = \{g_1, \ldots, g_{Cap_i}\}$ such that:

$$\forall C_i \in C, \exists g_k \in G, \alpha \in \mathbb{N}, g_k \times \alpha = i_i$$

- incremental IST updates
- heuristic: maximal GCD between two i_i

1: T: new registered interval
2: $G.append(T)$
3: **if** $|G| > Cap_i$ **then**
4: $g\_select \leftarrow g_i \in G, \forall g_k \in G, gcd(g_i, T) \geq gcd(g_k, T)$
5: $G.replace(g\_select, gcd(g\_select, T))$
6: **end if**



SNT

Outline

1 Introduction

2 CCN Background

3 Problem Definition

4 Solution

5 Evaluation

6 Conclusion

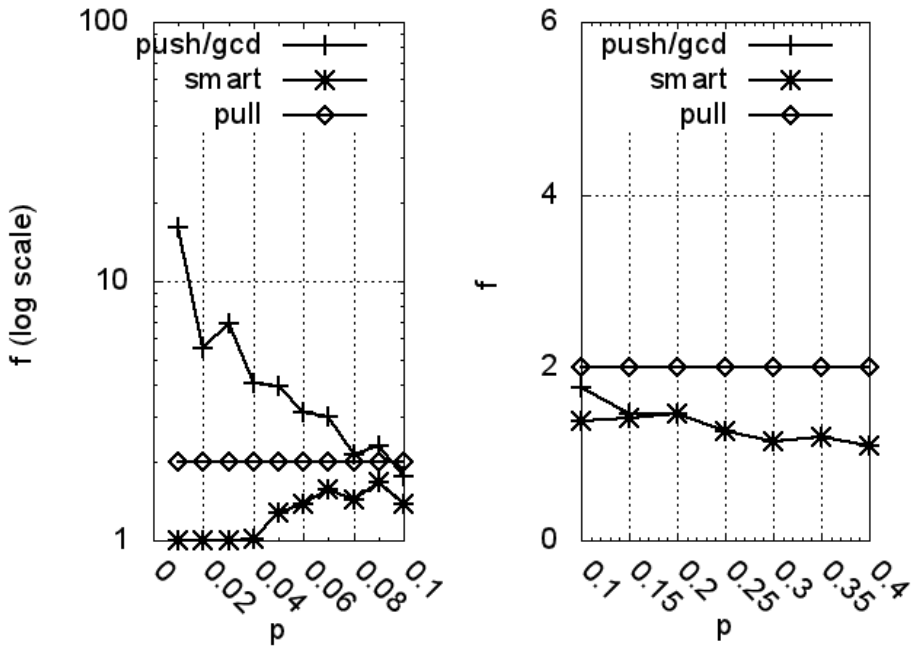
SNT

Methodology

- Basic simulator
  - limited functions necessary to assess our approach
  - no PIT or Content Store, no physical layer...
- Parameters
  - MAX_T: maximal sampling interval that a consumer can subscribe to,
  - p: number of consumers with a unique sampling interval expressed as a proportion of MAX_T
  - $\#counters$: number of available counters,
  - $\#steps = 5 \times MAX\_T$ the number of simulation step, 1 step = the initial sampling period of the sensor
- Strategies
  - $s \in \{pull, push, optimal, gcd, smart\}$
  - message overhead: $f_s = msg_s / msg_{optimal}$
- i_i chosen randomly
- simulation executed 10 times

Number of consumers

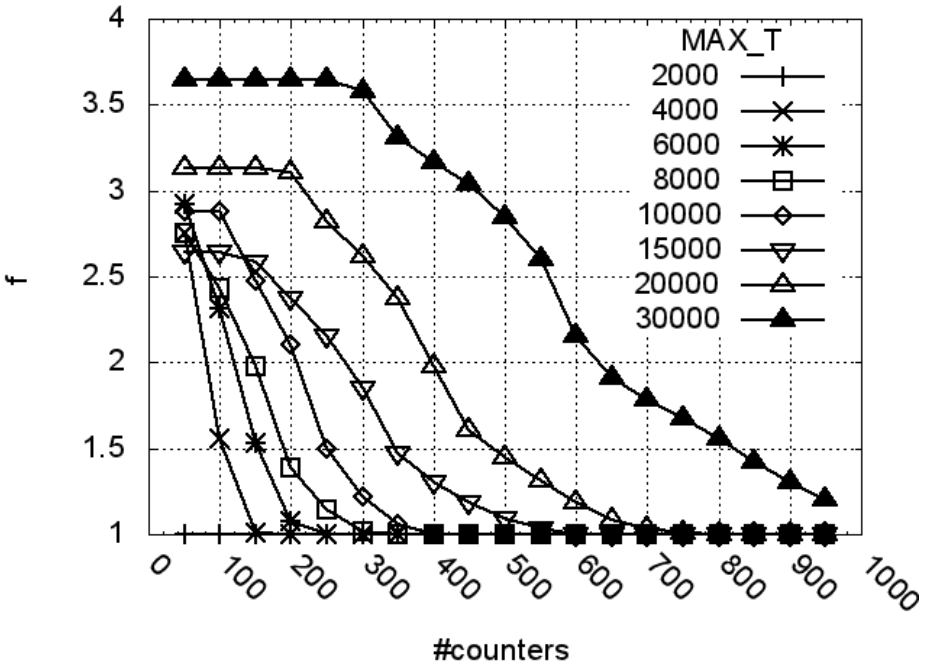
- $MAX\_T = 3000$, $\#counters = 100$
- *pull* strategy $\rightarrow$ 2 messages (1 interest + 1 data)
- prime numbers $\rightarrow$ $gcd = 1$ $\rightarrow$ $gcd \sim push$ (forwarding every packet)
- when p increases
  - more consumers $\rightarrow$ higher chance to have low i_i including 1
  - $\rightarrow$ all strategies tends to 1
- smart forwarding benefit
  - $1 < f_{smart} < 1.67$
  - $p < 0.04$ a unique counter per i_i (equivalent to *optimal*)
  - worst case: 240 unique subscribed periods ($p = 0.08$)



Number of consumers

- $p = 0.05 \times MAX\_T$
- smart forwarding strategy
- $\#counters$ and MAX_T vary

- when $\#counters$ increases $\rightarrow$ overhead reduces faster with less subscribers (lower MAX_T)
- fine tuning of Cap_i during the design phase of the sensor
  - assumption: a maximal overhead, a number of subscibers
  - $\rightarrow$ number of counters



SNT

Outline

1 Introduction

2 CCN Background

3 Problem Definition

4 Solution

5 Evaluation

6 Conclusion

SNT

Conclusion

- CCN for distributed IoT communication
- Context: producers (sensors) regulary updating consumers
- A new push based mechanism to limit the message overhead based on subscription requests
- Smart forwarding is close to the optimal forwarding and always better than the standard pull based communication of CCN
- Future work: multiple-content producers

Acknowledgement: This work was partly funded by IoT6 FP7 EU project under the grant agreement 288445.

CCN Traffic Optimization for IoT

Jérôme François, Thibault Cholez, Thomas Engel

UNIVERSITÉ DU LUXEMBOURG

SnT