

NetCodCCN: a Network Coding approach for Content-Centric Networks

J. Saltarin ¹ E. Bourtsoulatze ¹ N. Thomos ² T. Braun ¹

April 12, 2016

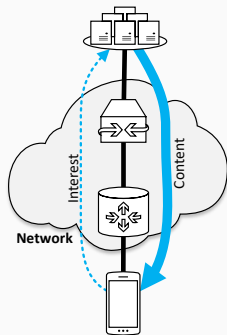
¹University of Bern, Switzerland

²University of Essex, United Kingdom

Introduction

CCN in a nutshell

1. Clients send Interest messages that contain the **name** of the content
2. Interest messages are routed based on their name
3. A node that holds a copy of the requested content replies to an Interest with a Content message

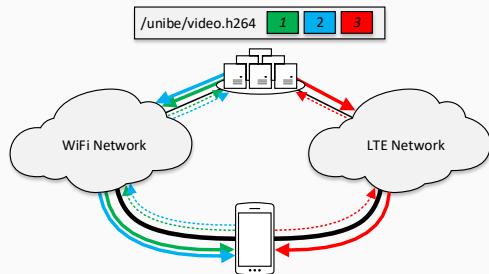


CCN: Handling large file request

The different segments that compose a large file can be requested in parallel over different interfaces.

Interests of the figure:

- `/unibe/video.h264/1`
- `/unibe/video.h264/2`
- `/unibe/video.h264/3`



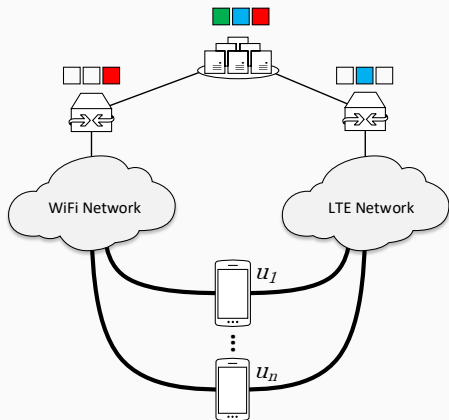
CCN: Non-optimal cases

Case1: Multi-cast

Clients must coordinate
Interest forwarding

Case 2: Multi-source

Client must know the location
of the segments



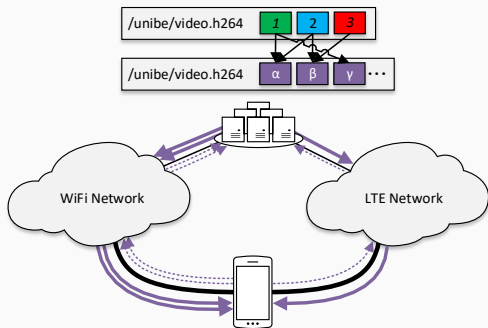
NC enabled CCN

Sources and routers perform coding operations on the segments.

Clients request coded segments, instead of specific segments.

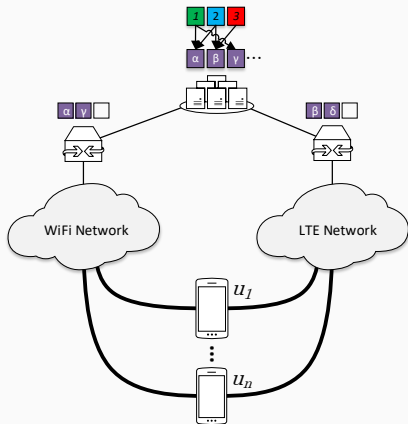
Interests of the figure:

- /unibe/video.h264
- /unibe/video.h264
- /unibe/video.h264



Benefits of NC enabled CCN

- Multi-cast, Multi-source issues alleviated
- Improved error resiliency
- Better throughput in networks with bottlenecks



NC enabled CCN: Name matching issues

Multiple Interests for different coded segments will carry the same name

- **Interest aggregation issue**

Interests for different coded segments are aggregated in the routers

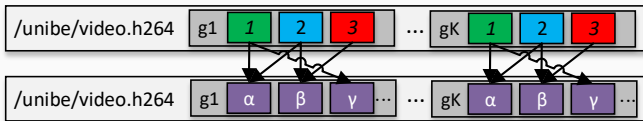
- **Caching issue**

The cache replies to all the Interests for different coded segments with the same segment

NetCodCCN

NetCodCCN: Content segmentation and naming

Content segmentation



Interest name

$\underbrace{\text{/unibe/video.h264/}}_{\text{file name}} \quad \underbrace{\text{g1}}_{\text{generation id}}$

Segment name

$\underbrace{\text{/unibe/video.h264/}}_{\text{file name}} \quad \underbrace{\text{g1}}_{\text{generation id}} \quad / \quad \underbrace{\text{110}}_{\text{NC header}}$

The **Interest aggregation issue** is solved by allowing the nodes to forward an Interest to its neighbor(s) only if if:

- The number σ_{fwd} of forwarded Interests is less than or equal to the number σ_{pend}^f of pending Interests on the face f where the Interest arrived

$$\sigma_{fwd} \leq \sigma_{pend}^f$$

The **caching issue** is solved by allowing nodes to reply to an Interest with a cached coded segment only if:

- **Case 1:** The generation is decoded, or
- **Case 2:** The number L of segments stored in the CS is larger than the number σ_{sent}^f of segments sent through the face f where the Interest arrived

$$L > \sigma_{sent}^f$$

Implementation & Evaluation

CCN codebase

- CCNx version 0.8.2

Network Simulation

- ns3 DCE (Direct Code Execution)

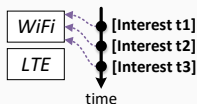
Metric

- Normalized delivery delay: $d = \Delta t_{measured} / \Delta t_{min}$

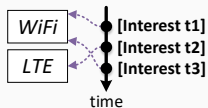
Implementation & Evaluation

Interest forwarding strategies

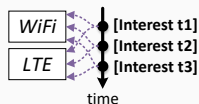
Default (DS)
Fastest face



Load Sharing (LS)
Distribute load over
all faces

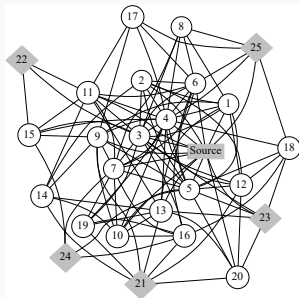


Parallel (PS)
All available faces



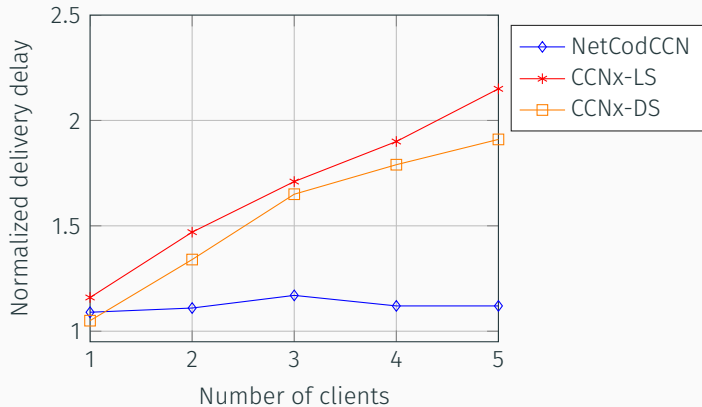
Evaluation: Set up

- Topology: random from PlanetLab
- Link capacity: 12Mbps
- Generation size: 100 segments
- Segment payload size: 5kB



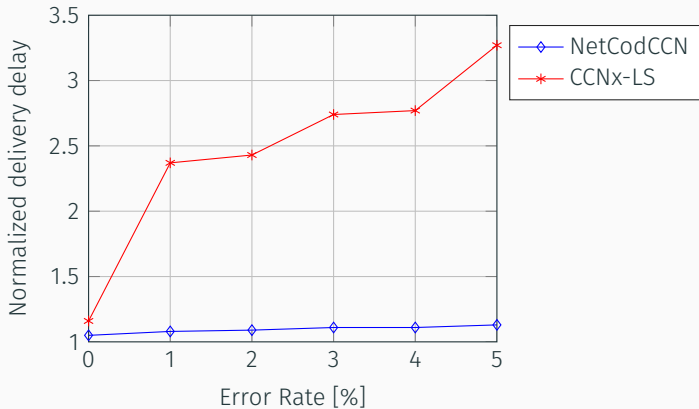
Evaluation: Number of clients

Normalized delivery delay versus the number of clients in the network



Evaluation: Error rate

Normalized delivery delay versus the error rate



Conclusion

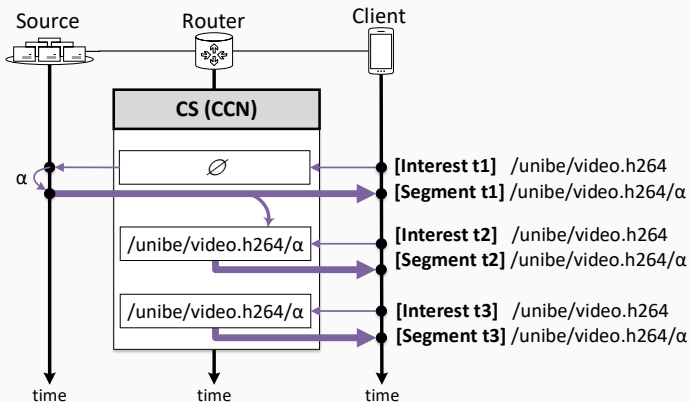
Conclusion

- Network Coding can improve content retrieval in CCN, in particular in multi-source, multi-client scenarios.
- NetCodCCN addresses the issues that network coding introduces to the original CCN.
- NetCodCCN can be further improved by optimizing the caching and the Interest forwarding strategies.

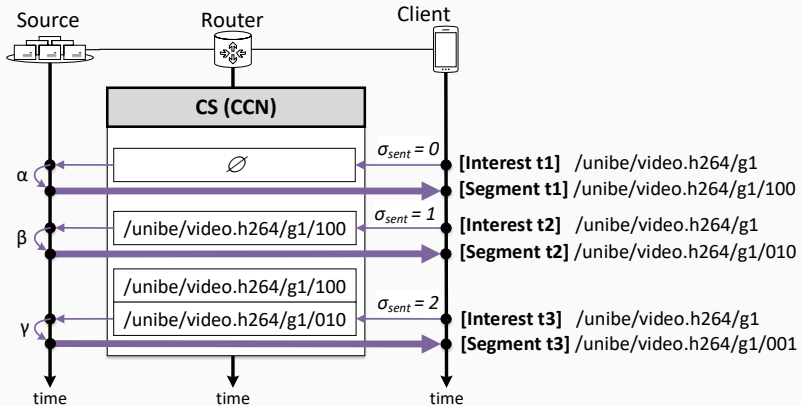
Questions?

Backup Slides

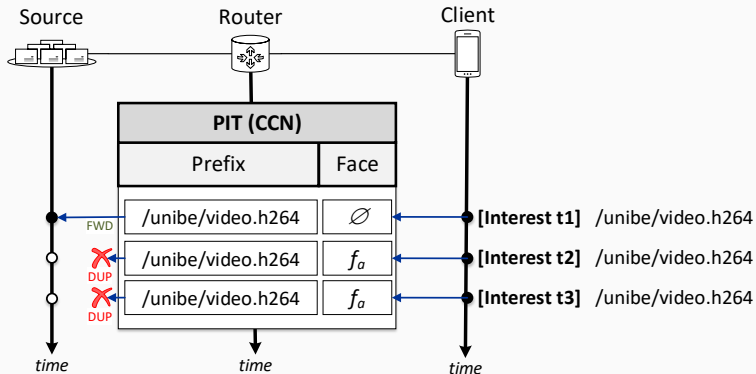
NC + CCN: Name matching issue (CS)



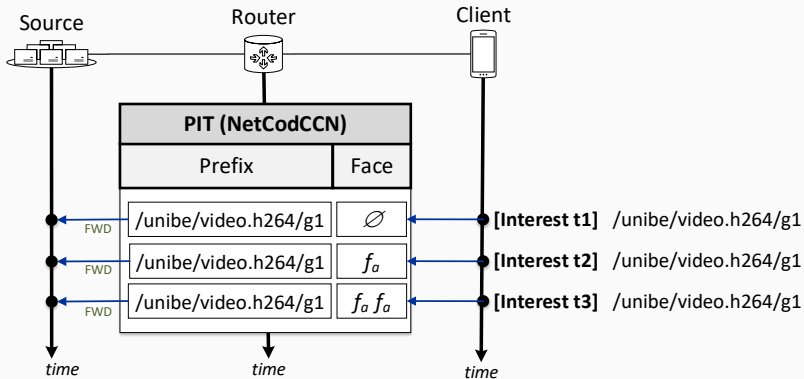
NetCodCCN: Name matching (CS)



NC + CCN: Name matching issue (PIT)

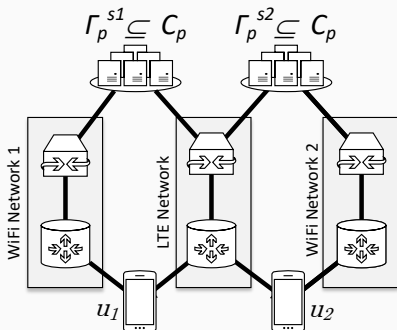


NetCodCCN: Name matching (PIT)



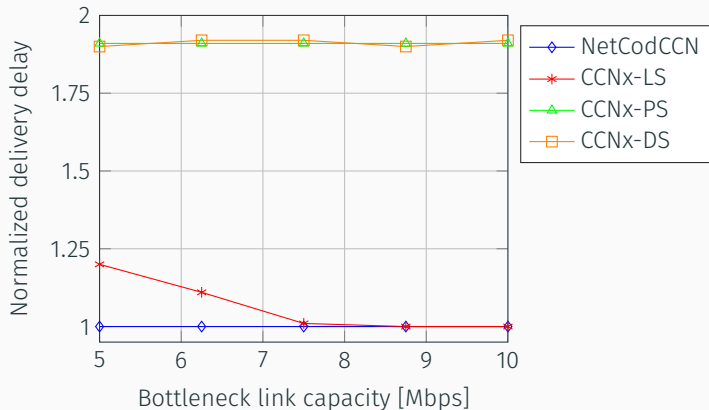
Evaluation/Butterfly topology/Set up

- Link Capacity: 5Mbps
- Generation size: 100 segments
- Segment payload size: 5kB



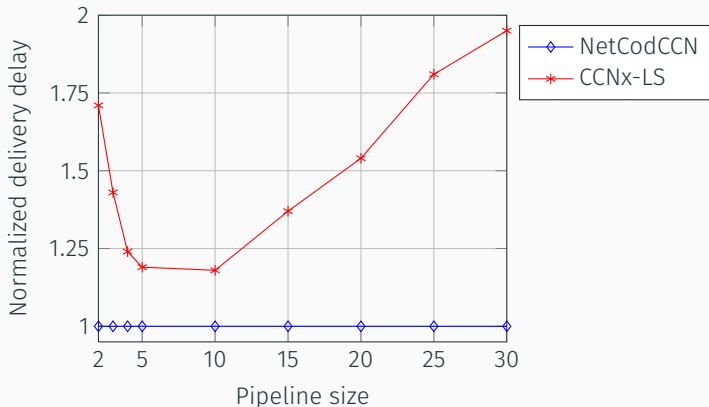
Evaluation/Butterfly topology/Results

Normalized delivery delay versus the capacity of the bottleneck link



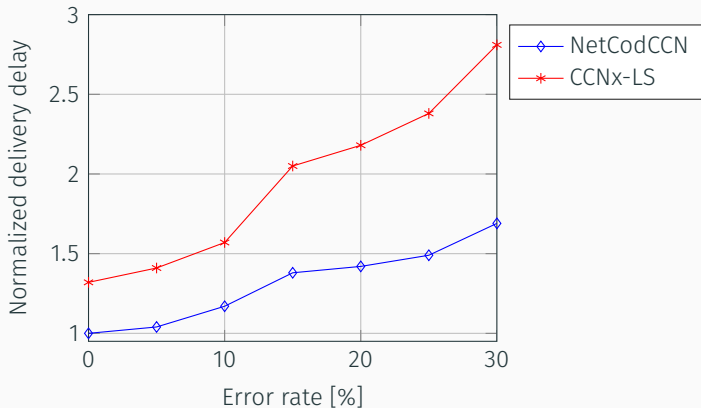
Evaluation/Butterfly topology/Results

Normalized delivery delay versus the pipeline size



Evaluation/Butterfly topology/Results

Normalized delivery delay versus the error rate



Evaluation/Butterfly topology/Results

Normalized delivery delay versus the source content duplication probability

