Adaptive Video Streaming over CCN with Network Coding for Seamless Mobility

Abinesh Ramakrishnan
Huawei
Cedric Westphal
Huawei and UCSC
Jonnhatan Saltarin
University of Bern

ICN & Network Coding

- RFC7933

> 9.5. Network Coding for Video Distribution in ICN
>
> An interesting research area for combining heterogeneous sources is to use network coding [Montpetit13b]. Network coding allows for asynchronous combining of multiple sources by having each of them send information that is not duplicated by the other but that can be combined to retrieve the video stream.
>
> However, this creates issues in ICN in terms of defining the proper rate adaptation for the video stream, securing the encoded data, caching the encoded data, timeliness of the encoded data, overhead of the network coding operations both in network resources and in added buffering delay, etc.
>
> Network coding has shown promise in reducing buffering events in unicast, multicast, and P2P settings. [Medard12] considers strategies using network coding to enhance QoE for multimedia communications. Network coding can be applied to multiple streams, but also within a single stream as an equivalent of a composable erasure code. Clearly, there is a need for further investigation of network coding in ICN, potentially as a topic of activity in the research group.

- M-J. Montpetit, C. Westphal, D. Trossen, Network Coding Meets Information-Centric Networking, in Proc. Workshop on Emerging Name-Oriented Mobile Networking (NoM'12), ACM MobiHoc, June 2012
- J. Saltarin, E. Bourtsoulatze, N. Thomos, T. Braun, NetCodCCN: a Network Coding approach for Content Centric Networks, IEEE Infocom 2016
- K. Matsuzono, H. Asaeda, T. Turletti, Low Latency Low Loss Streaming using In-Network Coding and Caching, IEEE Infocom 2017

Seamless Mobility Support

- **Heterogeneous interfaces…**
  - 5G, 4G/LTE, WiFi
- **…connecting to heterogeneous access points…**
  - Macro, Pico, WiFi AP, etc
- **…under wide ranging mobility conditions**
  - static, pedestrian, vehicular

- **Can network architecture support this heterogeneity efficiently?**

- **One issue of the current IP architecture: data transfer is severely constrained due to transport layer session**

Data transfer in the IP world

- **Data transfer is set up as follows:**
  1. **Map content name/URL to content hosting server**
  2. **Set up TCP/transport session between current location and server**
  3. **Transfer data**
  4. **Upon mobility event, go back to 2), using the location of 1), which may be suboptimal at this point**
- **The transfer is tied up to the initial host location…**
  - which may or may not be appropriate after mobility event
- **… the transfer is embedded within a session**
  - which has to be re-established upon mobility event,
- **…and the transfer is strictly point-to-point**
  - MP-TCP may allow multiple paths in between the two end points

Data transfer in ICN

- **Compare and contrast with IP… In ICN:**

  1.Request content by name
  - The network locates a copy, potentially on a nearby server

  2.Upon mobility event, request content by name
  - The network locates another copy, potentially on a nearby server

- **There is no session to carry over!**
- **This is still a point-to-point data transfer, and does not take advantage of the multiple interfaces on the mobile device**

- **Huawei’s solution: NCICN (network coded ICN)**

Seamless Mobility in 5G: NcIcn

- **Basic idea (from "Network coding meets ICN," ACM Mobihoc workshop, 2012):**
  - The mobile device sends interest for network coded chunks to all its
  - interfaces
    - The interest is not for a specific chunk, but for one in a subset of all chunks
  - The interests are forwarded to a node that holds the content
  - It responds with a random linear combination of the chunks in the requested range
  - Once the mobile device has enough network coded chunks (NCCs), it decodes them to retrieve the original content
  - Intermediate node may or may not cache the NCCs
    - They may hold onto them until they can decode them, for instance
- **It creates a point-to-multipoint connection over the several interfaces**

Why?

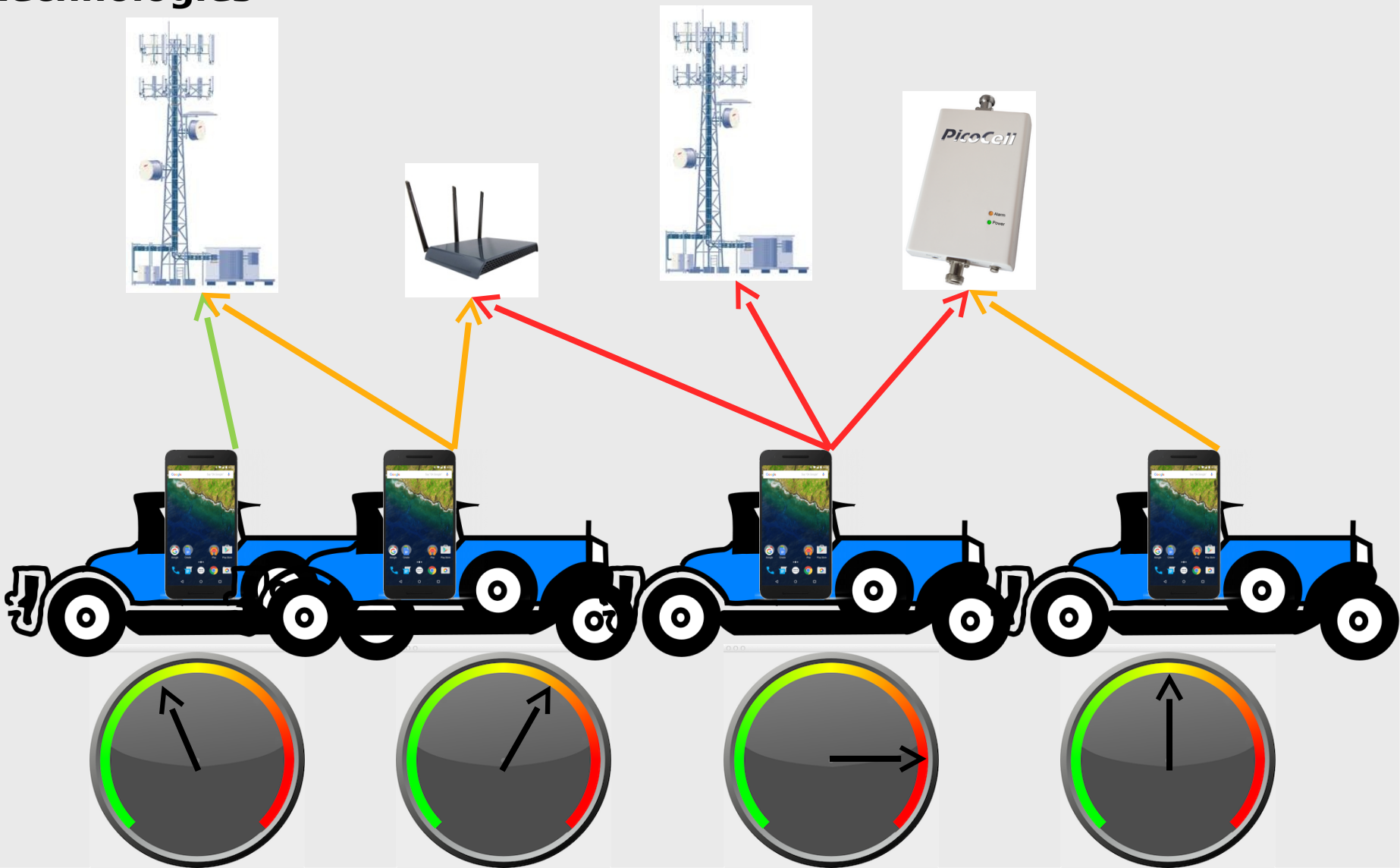
- **Several benefits from NCICN**
  - It builds atop the session-less nature of CCN/NDN
    - No need to establish a TCP session, no need to identify a specific server
  - It allows point-to-multipoint data transfer
    - The content can be requested from multiple interfaces *at the same time*
    - Because the returned content is NCCs, it receives new information on each of the
    - interfaces
  - But it does not require *any synchronization* between these interfaces*!*
    - Each path may have an unknown, varying bandwidth/latency, they all add up
  - It forms a logical link between the mobile device and the content over
  - multiple locations
    - The link capacities of each interface add up
    - Rate adaptation for video streaming can work on the aggregate capacity, no need
    - for fine grained link-by-link rate adaptation!
  - It populates the cache faster (if interests are forked at a router and return multiple responses, only one is sent back to the mobile device, but all can be
  - cached)
  - Small or no performance loss in the point-to-point case; large gain in other

Rate Adaptation Logic

- **The video client only sees an aggregate performance.**
- **It does not know what underlying link there is, but with network coding + CCN, the link add up and the performance is abstracted**
- **Rate adaptation can be performed upon the aggregate bandwidth with no modification in the client**
- **The NCCs are created so that each generation of NCCs coincides with a segment at a given rate**
  - This allow the rate adaption with network coding
  - This minimizes the delay of network coding, since the
  - video client never waits for more than one segment to
  - be decoded
- **NCC can additionally take advantage of caching enabled by CCN (not discussed here)**

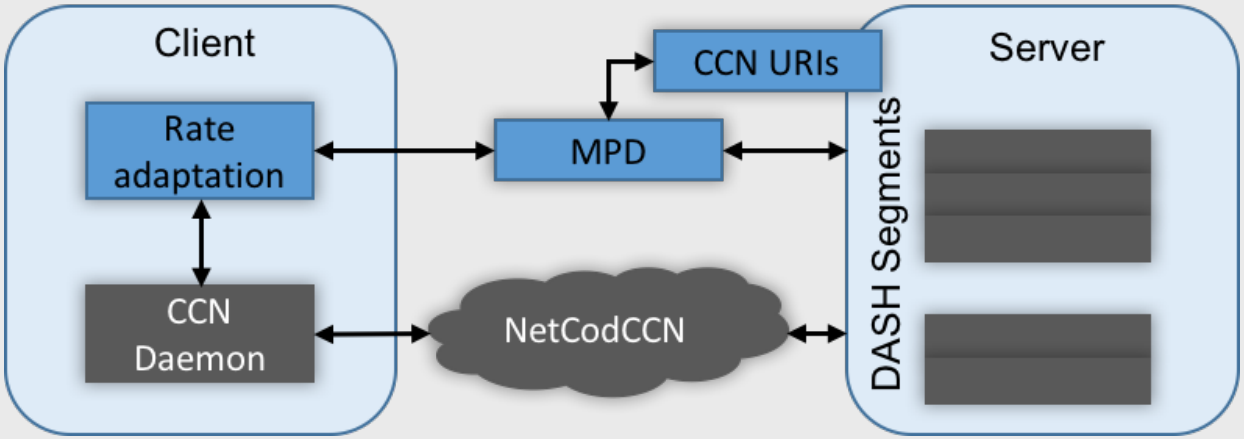
Basic Use Case

- **Node seamless retrieve the content from heterogeneous access technologies**

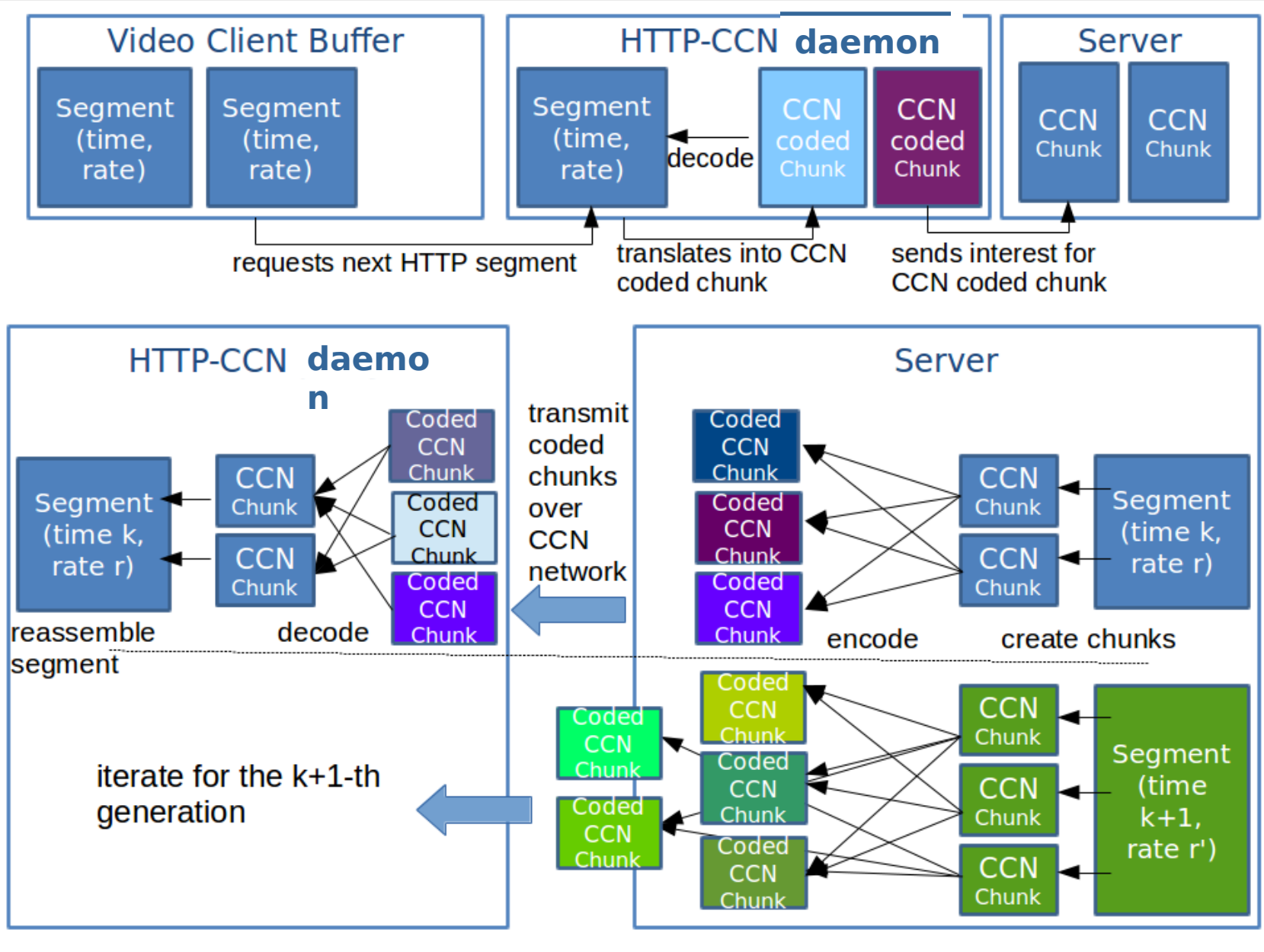


Encoding Method

- **Each DASH segment for a time step and a given rate is chunked into CCN packets: they define a generation**
- **The packets of *the same* generation are encoded together**
- **The interest name for any chunk of a given generation is the same**
- **The network coded packet carries a header that describes the linear combination of the original chunks of the generation**



Encoding Method

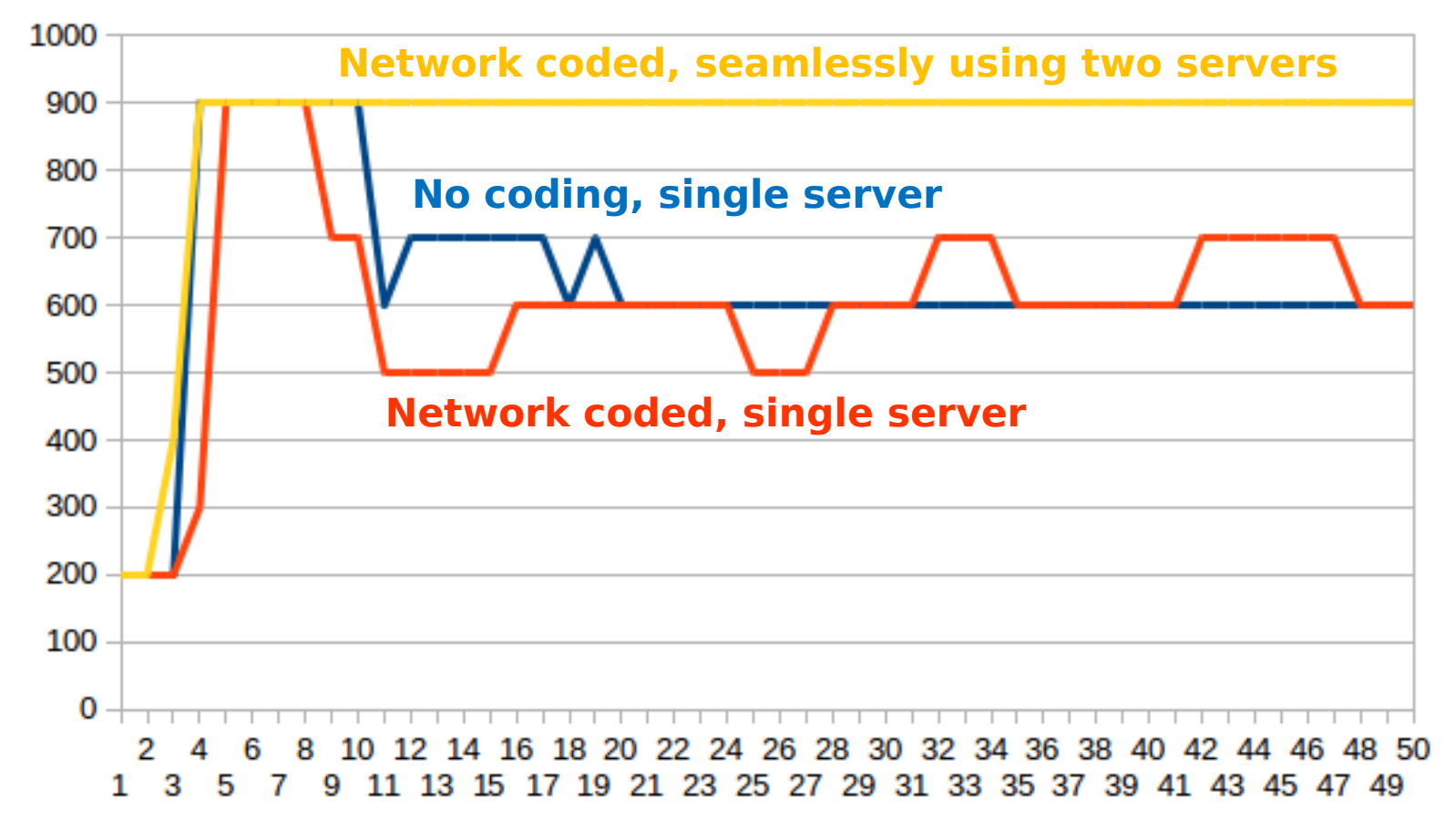


Implementation

- **Currently implemented in the lab**
  - Client connected to 2 servers through Ethernet cables
  - Servers and clients in VMs, connected with CCN networks in between
  - Cables can be plugged in and out, mimicking the mobility events
  - Video streaming application adapts to the underlying capacity of the
  - aggregate links
- **Next step**
  - Move from CCN 0.8 to NDN/CCN1.x (issue with video client VLC that only supports older version of CCN)
  - Car driving over the parking lot, connecting to several APs, while streaming a video (and take customers for a ride)

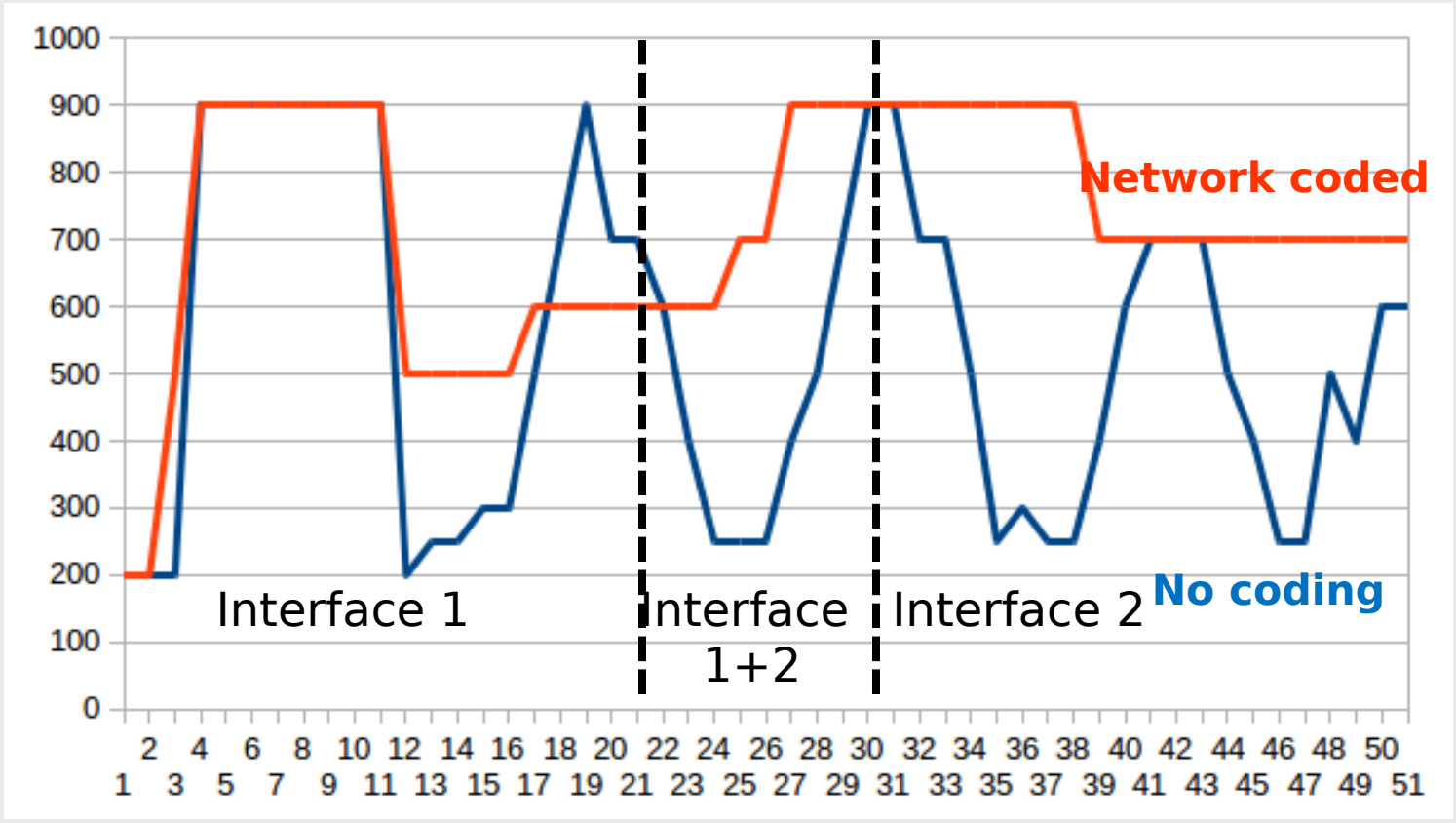
Lab Results I

- **Rate adaptation evaluation: static scenario**



Lab Results II

- **Rate adaptation evaluation: dynamic scenario**



What's Next?

- On Thursday, discussion on network coding and ICN in the IRTF NWCRG

- Feedback? Worth formalizing further for CCN or NDN?