Named Data Networking for Social Network Content delivery

draft-truong-icnrg-ndn-osn-00.txt
IETF 92, ICN RG, March 24, 2015

P. Truong, B. Mathieu (Orange Labs), K. Satzke (Alu)
E. Stephan (Orange Labs)

Timeline

- eCousin Project
- Online Social Networks (OSNs) analysis
  - locality observation
  - optimization simulation
- NDN-based network architecture with local routing for OSN content delivery
- SDN-based routing configuration employing OSN information

Project eCOUSIN

ecousin-factsheet.pdf eCOUSIN

- http://cordis.europa.eu/fp7/ict/future-networks/documents/call8-projects/

- eCOUSIN
  - eCOUSIN: Aims at designing a service aware network architecture with built-in https://ecousin.cms.orange-labs.fr/

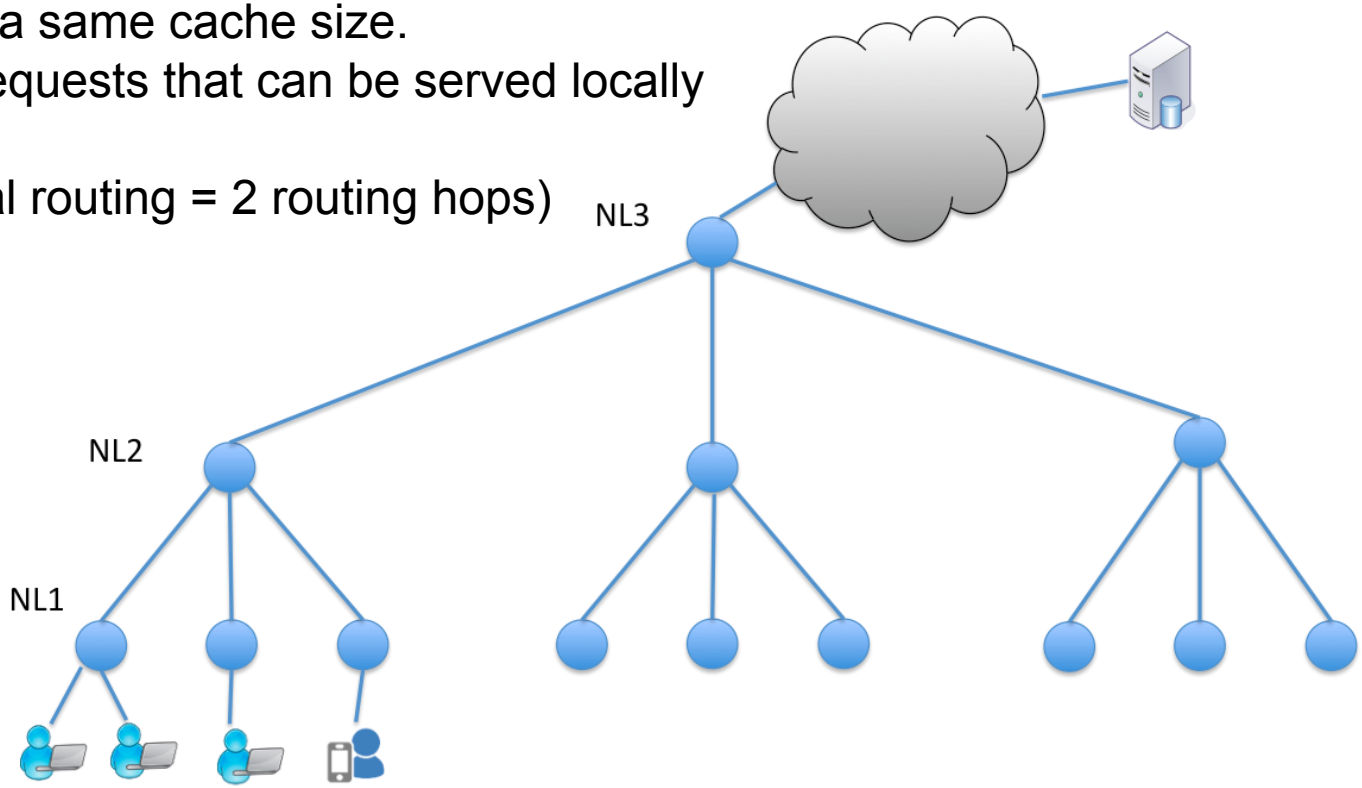
improve the efficiency of content delivery.

- From traffic analysis we observed 2 things:
  - the local end-users' behaviour
    - majority of users are frequently connected to users of the same area, e.g. in the same access ISP.
    - Only very popular users have friends distributed everywhere.
    - A majority of contents are delivered to local users, except for very popular accounts.
  - the network distance between end-users
    - The networking path computation does not take in account the network distance between end-users
    - Local End-users' requests are directed toward remote OSN servers or CDN caches, generally outside the access ISP network
- Conclusion :
  - The ISP would save transit cost and improve end-users QoE if it could route the OSN traffic exchange between local users.
- Directions of study:
  - Usage of users relationships in NDN (Named Data Networking) to route the delivery of OSN content

From traffic analysis we observed 2 things:
majority of users are frequently
, e.g. in the same access ISP.
remote OSN ISP
network
Conclusion :
if it could
route the OSN traffic exchange between local users.
Directions of study:
(Named Data Networking) to

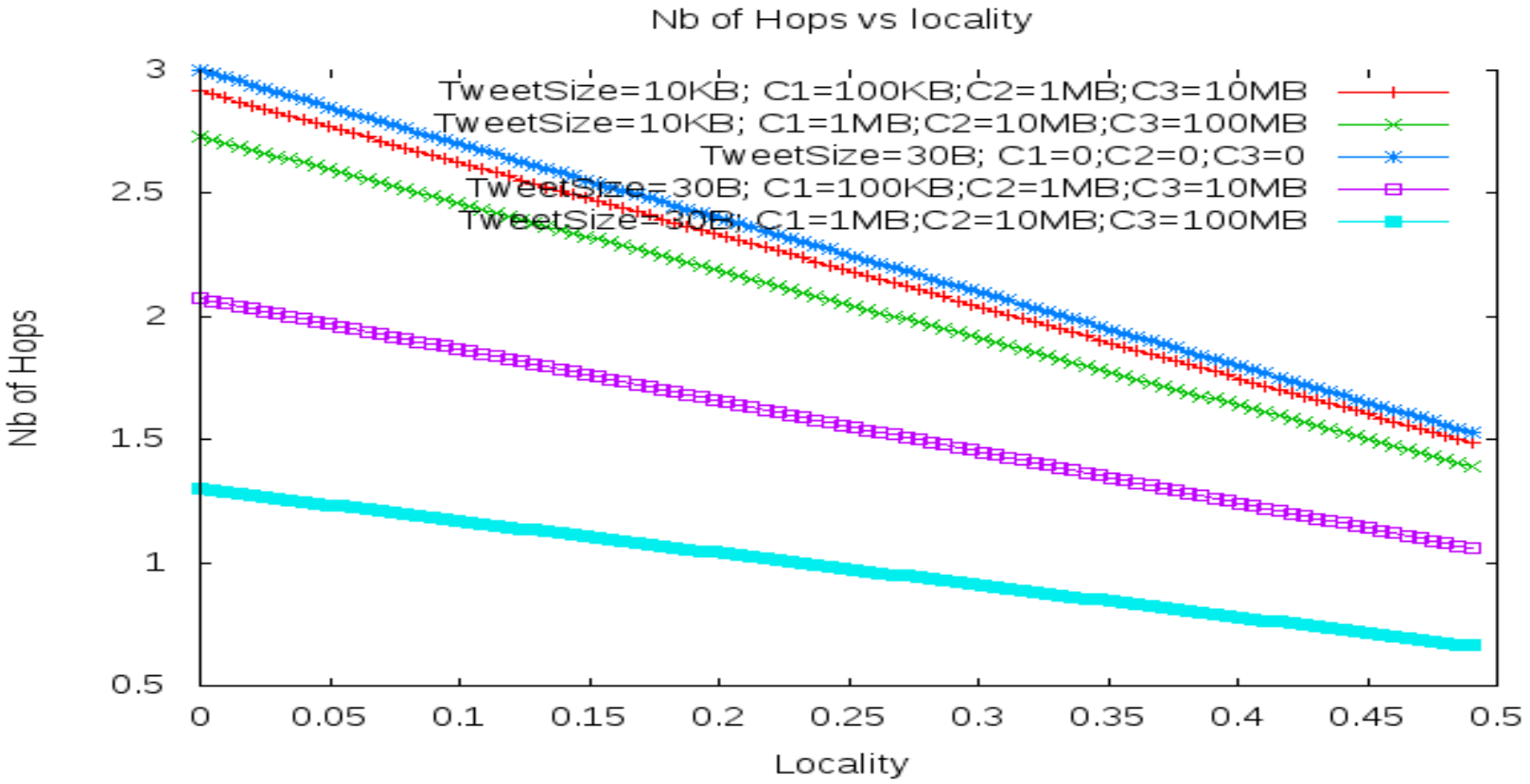
Modeling Local-aware NDN-based Routing

- In our simulation we considered :
  - network topology limited to 3 levels
  - *E*nd-users request content of same size,
  - Content size is from 10k to 30k bytes,
- At a network level, all the routers have a same cache miss probability and a same cache size.
- *W* = ratio of users' requests that can be served locally in the network.
- local regions for local routing = 2 routing hops)

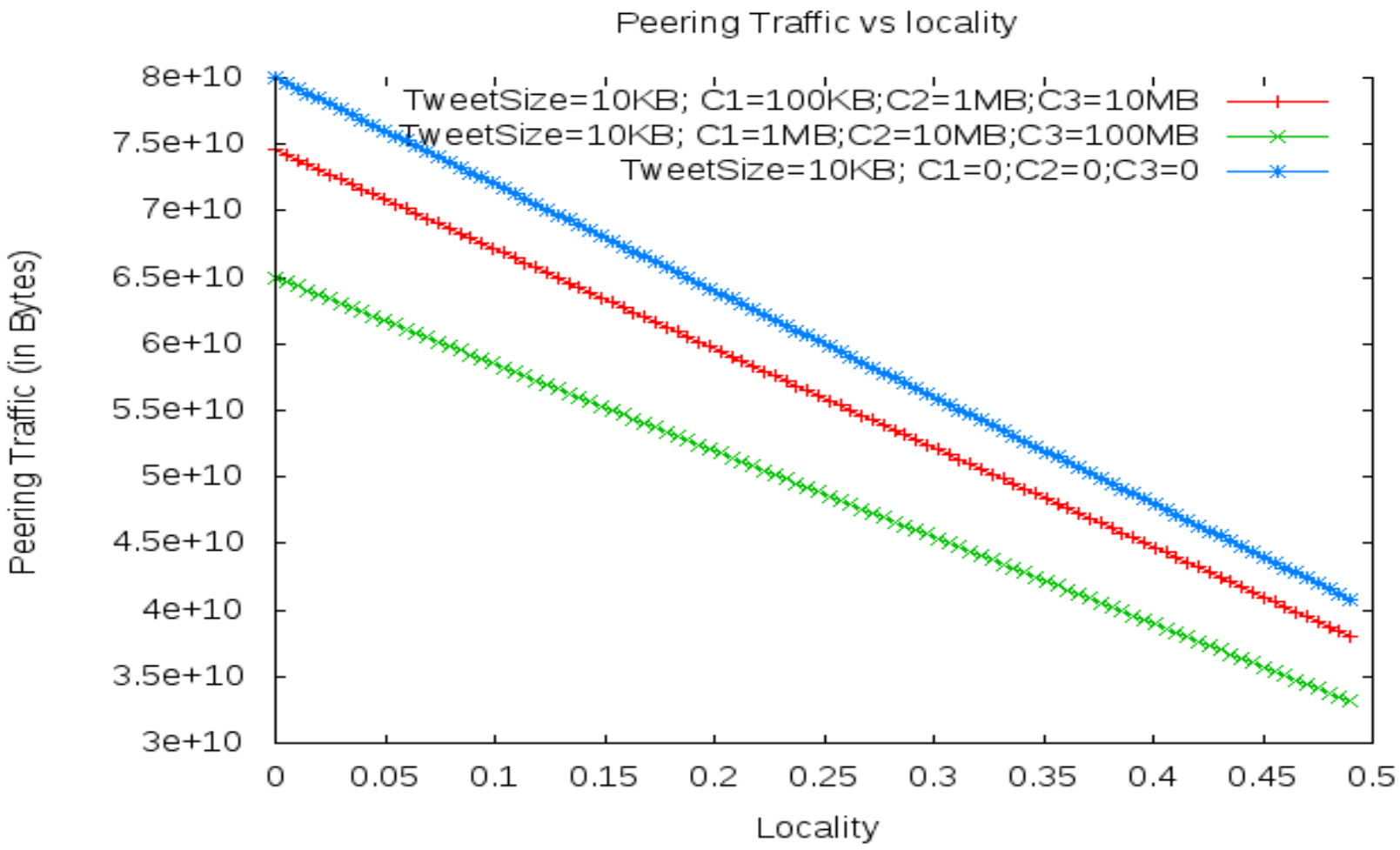


Analytical Evaluation:
Number of Routing Hops vs Locality

- Locality has a huge impact on the delivery**: the number of routing hops can be largely reduced (up to 60%)**, also when no cache is used.



Analytical Evaluation: Peering Traffic vs Locality

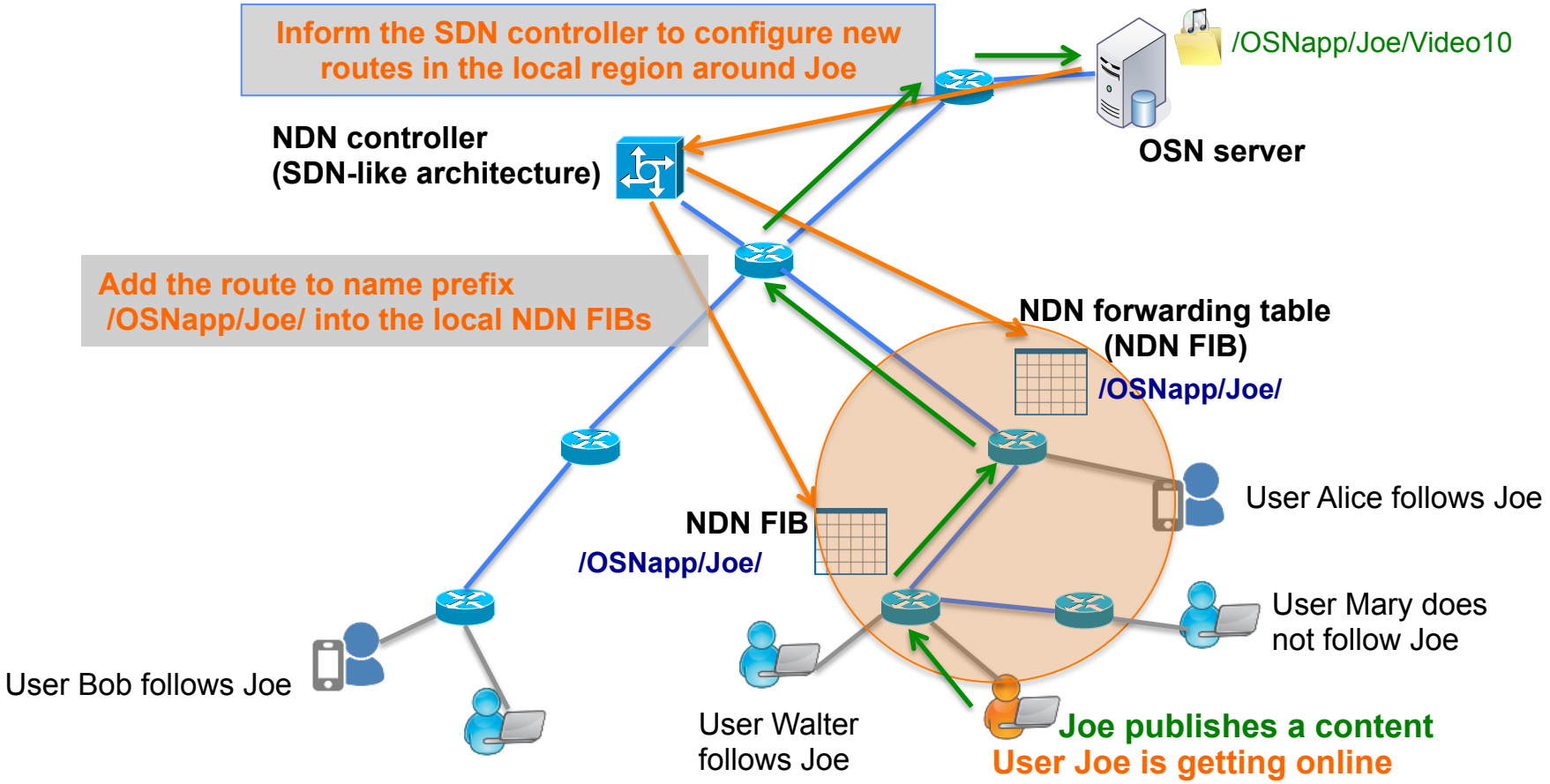


Improving Content Delivery in OSNs with Local-aware NDN-based Routing

- We use the NDN architecture to **optimize the networking behaviour while better reflecting the local end-users behaviour**
- Popular end-users, whose content is consumed worldwide, should have a different way of working than non-popular local end-users, whose content will be locally consumed
- We perform **local routing between the end-users who are in the immediate vicinity**.
- **Locality** is defined by network routing hop:
  - Two users are local if there are separated by 2 routing hops (or any other value depending on the design configuration).
- A **centralized (SDN-based) controller** allows to dynamically **configure the NDN forwarding tables, based on the social interactions in the OSN**:
  - e.g. add route in the local NDN routers if close friends/followers are on-line, remove it if not.

Publication of a Content

- For naming end-users and their contents (text messages, videos, photos, etc), we suggest the following hierarchical naming:
  - /OSNapp/userID/contentID
- For local routing, we consider local network regions of 2 routing hops.

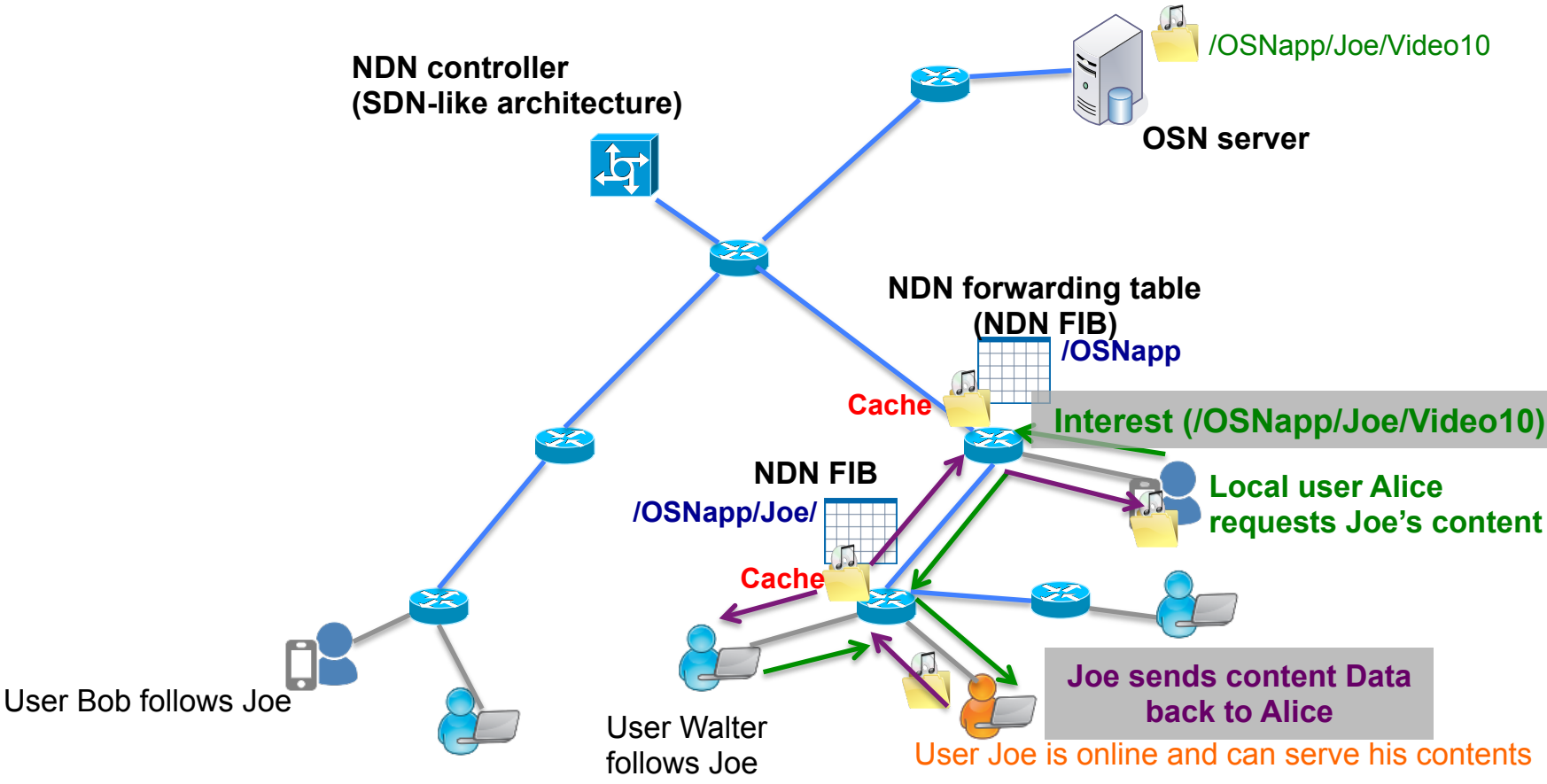


Publication of a Content (cont.)

- Joe is getting online: he connects to the OSN server using the application client on his smartphone or computer.
- Joe wants to publish a content
- The OSN server informs the NDN controller with a notification message containing information related to Joe:
  - the location (NDN name) of the NDN router to which Joe is connected
  - the list of Joe's friends/followers who are currently online, along with their location (i.e. the name of the NDN routers to which Joe's online friends are connected)
- The NDN controller configures the local NDN routers (located at most 2 routing hops from Joe) by adding the route "/OSNapp/Joe" for Joe reachability in the related FIBs.
  - **this route is added to a local NDN router only if there are Joe's online friends attached to the router.**
- Joe can now publish his new content: the content object is stored in the OSN server.

Local Routing for Retrieving Content from Local Users

- Local users (2 routing hops far away from Joe) will get content directly from Joe, instead from the OSN server



Local Routing for Retrieving Content from Local Users (cont.)

- Local user Alice wants to retrieve Joe's content. She then express an Interest for "/OSNapp/Joe/Video10".
- Thanks to the previous routing configuration by the controller, the Interest("/OSNapp/Joe/Video10") will be routed to Joe.
- Joe's OSN app client, receiving the Interest, issues a Data message to Alice for serving the requested content Video10.
- While forwarding the Data message on the reserve path (taken by the Interest), Joe's content is cached into the Content Store of all the traversed routers.
  - Optimization: We keep the content in the cache only if there more than N users potentially interested by the content
- If now Walter wants the same content, he will get it from the cache

SDN-like Routing Configuration Employing OSN Information

- The logically **centralized controller** allows to:
  - **set up the default route "/OSNapp"** in NDN routers to forward requests to the OSN server if no other longer prefix match is found in the FIB.
  - **dynamically configure** the FIB tables for **local routing between users.**

- The controller configures routing **based on social information provided by the OSN server:**
  - when an user is getting online, server notifies controller to add routes for local routing to local NDN routers where there are user's friends;
  - when the user is disconnected, server notifies controller to remove routes;
  - controller can add routes in local NDN routers according to the network operator's requirements and policies for a certain network region and/or a particular group of contents/users, etc.

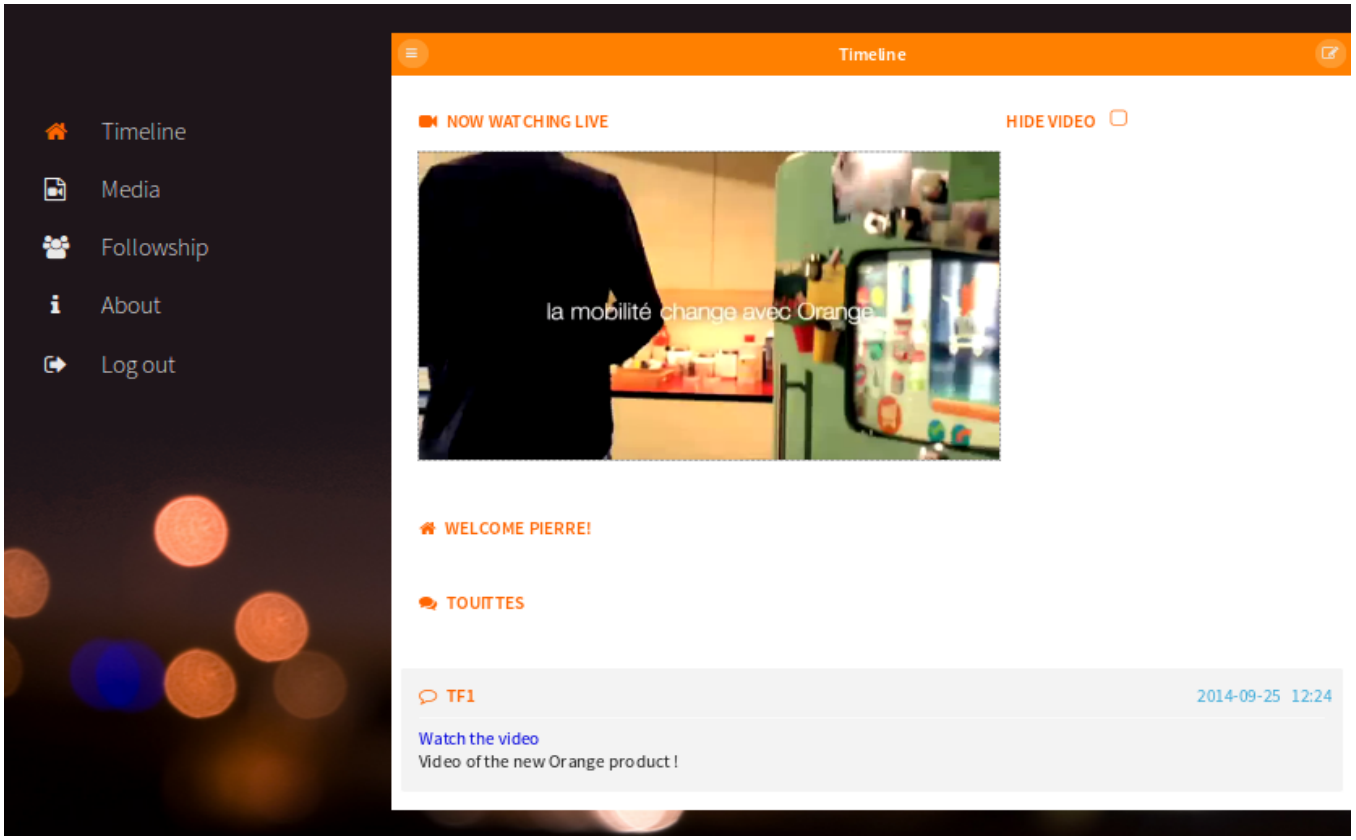
Social-aware Optimization of FIB for Scalability

- The OSN server instructs the NDN controller to **modify the FIB for local routing** to the only NDN local routers where the user's **number of followers exceeds the threshold value**. [number] is defined to locally **quantify users' popularity**.
  - The user is getting online, the server notifies the controller for routing updates only if the user wants to publish content
  - If a **user is inactive for a long time**, the server can ask the controller to **remove the routes** related to this user;
    – If the user is getting active, the server can notify the controller to add routes again.
    - **Space/Time-sharing activity**: limit the number of local end-users' routes in a FIB
      – by selecting the most popular end-users;
      – or using a round robin rule : one route is enabled only for a limited period of time, and needs to wait for some moment before getting active again.

routes in a FIB
– by selecting the most popular end-users;
– or using a round robin rule : one route is enabled only for a limited period of time, and needs to wait for some moment before getting active again.

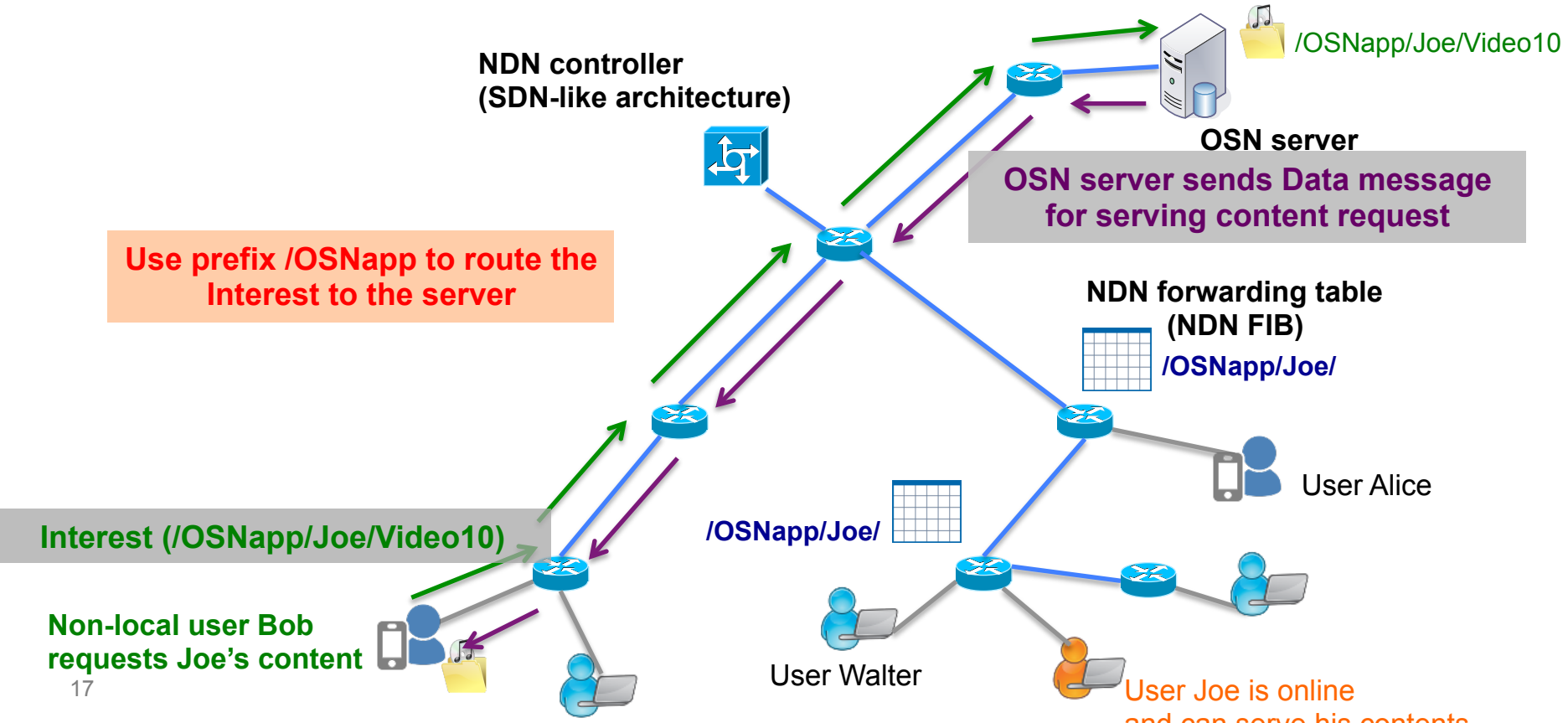
Thanks. - Q&A

NDN Implementation



Using OSN Servers to Retrieve Content for Non-Local Users' Requests

- Non-local users will be served by the OSN server



Using OSN Server to Retrieve Content for Non-Local Users' Requests (cont.)

- Non-local user Bob wants to retrieve Joe's content. He sends an Interest for "/OSNapp/Joe/Video10".
- As Bob is a non-local user, i.e. located too far from Joe, his Interest will be forwarded using the default route "/OSNapp".
- The OSN server is responsible for serving the request:
  - The sent content will be cached in the content store of the different traversed NDN routers on the reverse path.

NDN (Named Data Networking)

- Content objects are published, discovered, retrieved and forwarded based on , no longer on endpoint addresses

- End-user sends an **Interest message** for requesting a content object based on its name
- This Interest message is **routed/ identified by the name** of the requested content object.

- A **Data message** containing the content object is returned back to the end user along the reverse path as a response.
- This Data packet is also identified by the same content name. This allow to ”**cache**” the content object in the traversed NDN nodes.