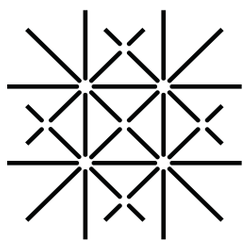
Towards a Monitoring Protocol over Information-Centric Networks

Dima Mansour, Christian Tschudin
dima.mansour@unibas.ch



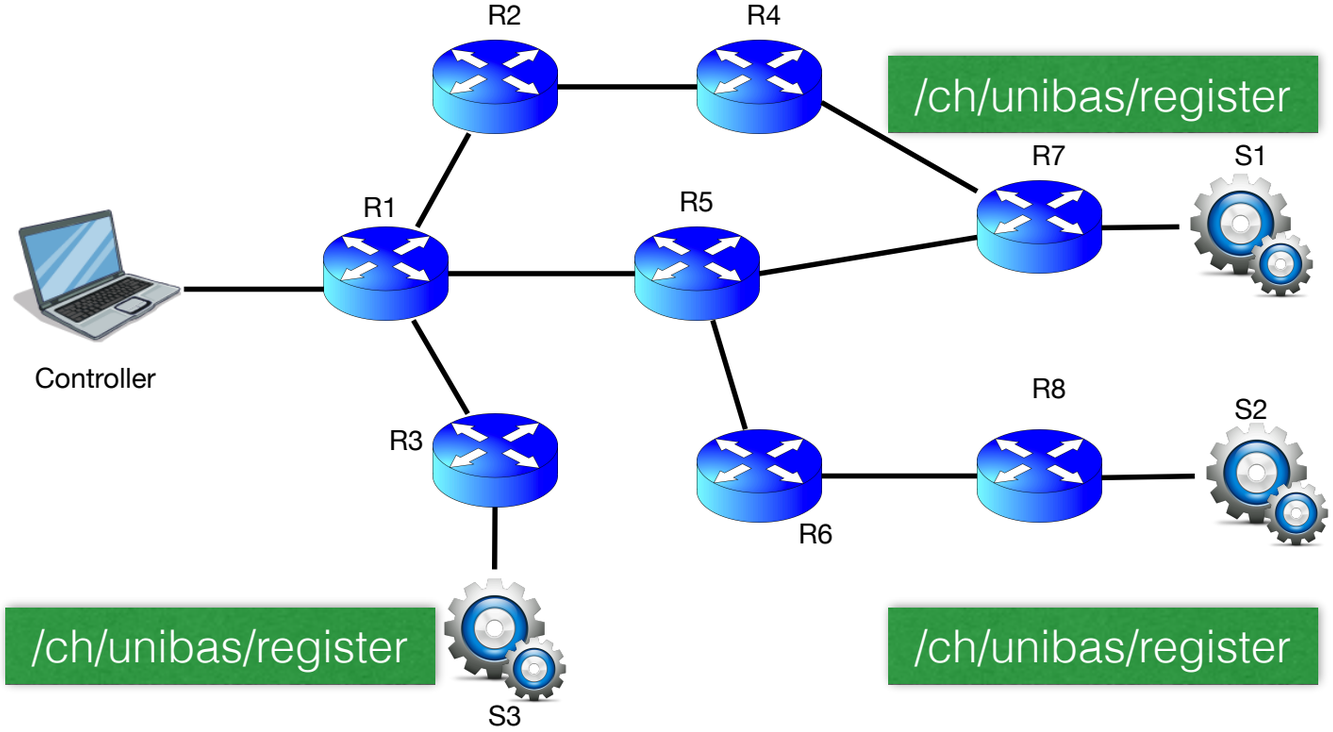
Universität Basel

27- 09 - 2016

What if… there is not one repo but several of them for the same content/service ?

In other words: How to undo ICN's de-localization of replicas?

Once this is possible, we can: count the replicas, monitor individual instances, reconfigure each of them, control load-balancing …



Services are invoked by name
/ch/unibas/register,
/ch/unibas/repo/XYZ
…
over ICN

Overall goal: *is to manage replicas (repos, services …)*

Overview

- Traditional monitoring protocols
- Challenges of ICN monitoring
- Assumptions
- Name-Centric Monitoring Protocol (NCMP)
- Discussion
- Summary and future work

Traditional Monitoring Protocols

- IP-based network - mgmt starts with an IP addr, or broadcast
- predefined ports of data/services

- Retrieving information from network entities
- Receiving alerts (aka SNMP traps)
- Discovering failure of nodes
- Remotely resetting or reconfiguring a device

How to do this in ICN?

Challenges of ICN monitoring

- How can we collect information about the machines while we only now de-localized names?
- How can we know which reply belongs to which replica?
- How can we discover failure of replicas? (—> monitoring)
- How can we distribute the load between replicas?
- How to reset/ reconfigure/ initiate a replica

Traditional monitoring protocols over IP-based data center cannot be easily applied on ICN-based data centers

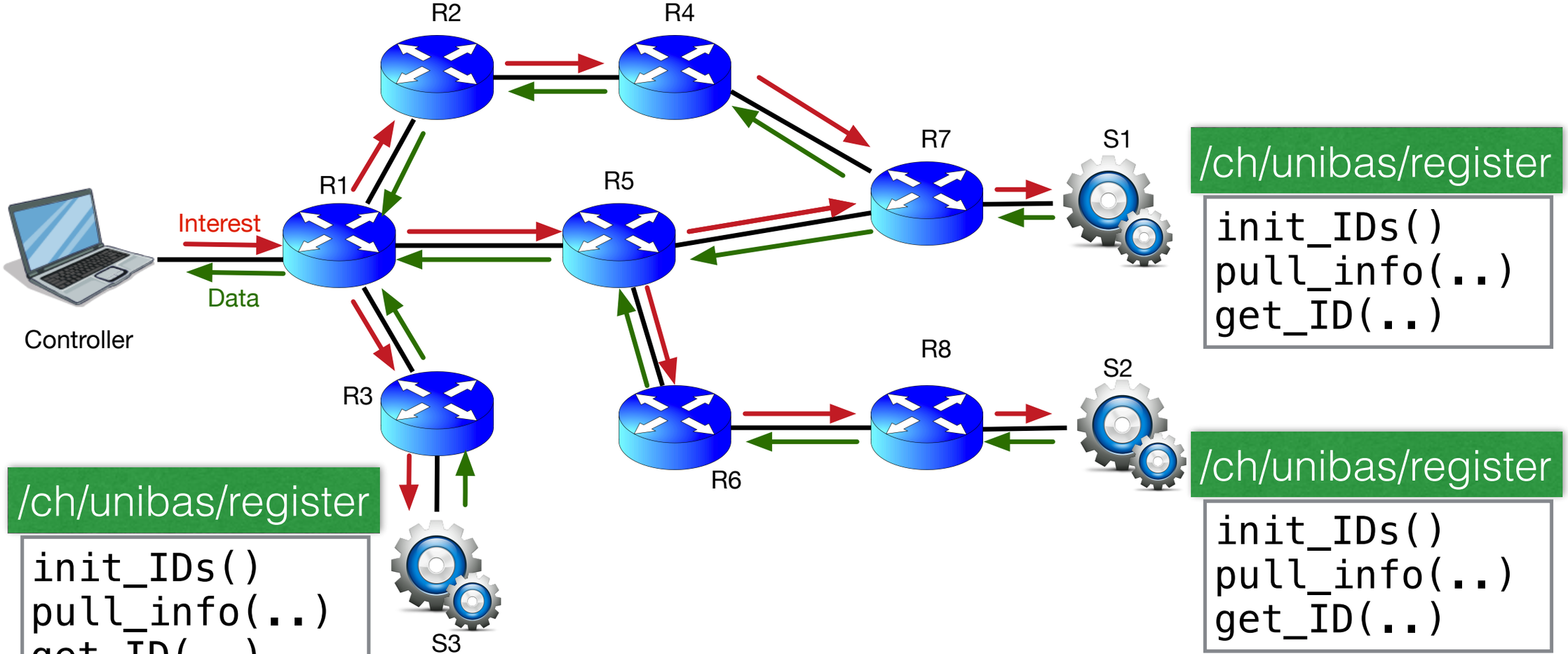
Assumptions

- A pure ICN (no underlay network, RPC, other magic)
- A dedicated *controller* machine
- The controller has no previous knowledge about the number of replicas or were the replicas are installed
- The controller deals only with service names.

*How can the controller **scan** different replicas serving the same named entity to **collect** run-time information from them?*

Name-Centric Monitoring Protocol ***(NCMP)***

- NCMP allows the controller to connect with all replicas in pure ICN manner
- NCMP has three published service end points:
  `init_IDs(), pull_info(…), get_ID(…)`
- *NCMP* relies on a broadcast forwarding strategy



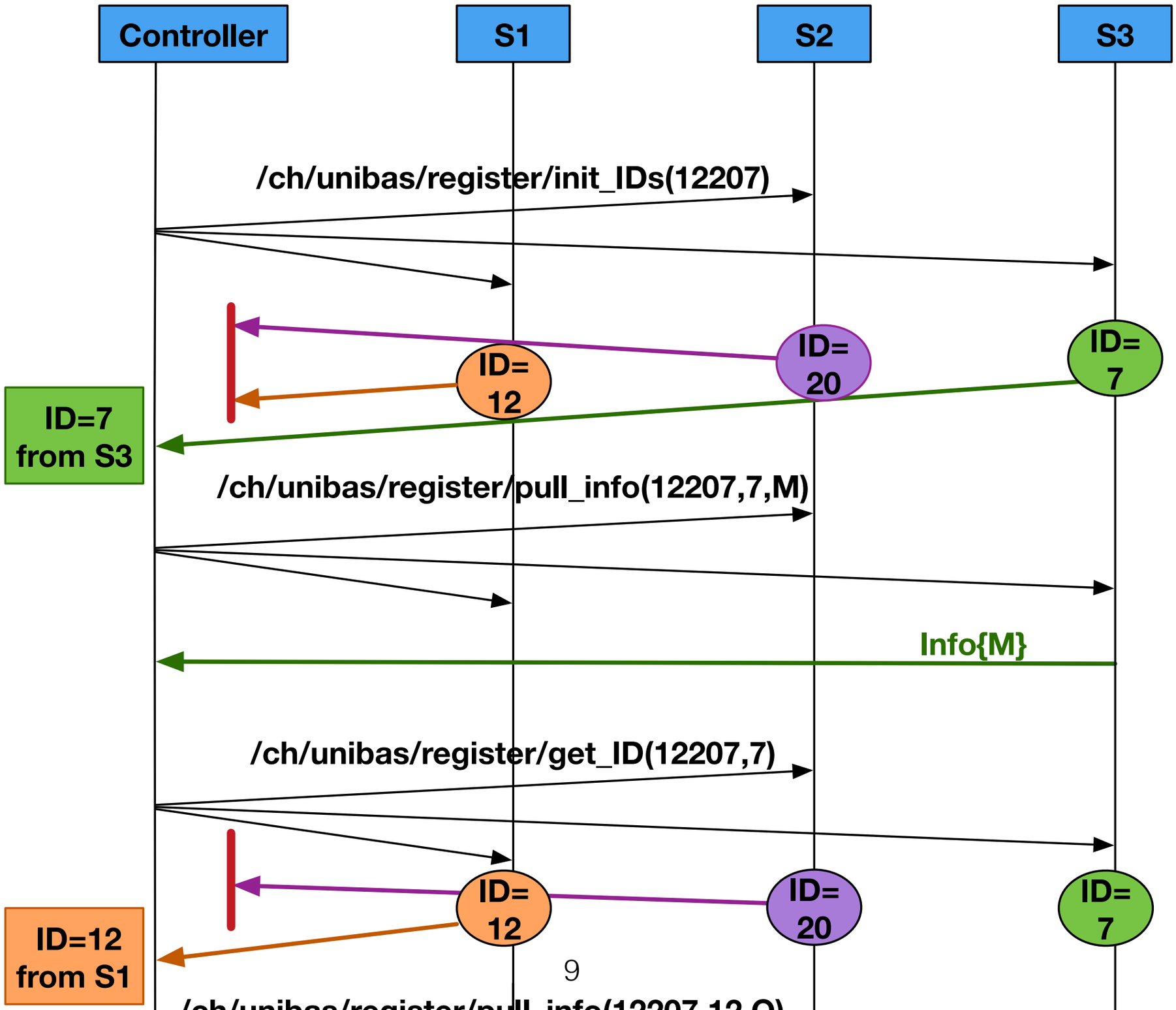
NCMP service end-points

- `init_IDs(roundID)` *:*
  assign yourself a new ID

- `pull_info(roundID, ID, propertyName)` *:*
  Return requested property if ID matches

- `get_ID(roundID, excludeList)` *:*
  only reply if your ID is *not* in the list

Name-Centric Monitoring Protocol ***(NCMP)***



/ch/unibas/register/get_ID(12207,7)

ID=12 from S1

ID= 12

ID= 20

ID= 7

/ch/unibas/register/pull_info(12207,12,Q)

Info{Q}

/ch/unibas/register/get_ID(12207,7,12)

ID=20 from S3

ID= 12

ID= 20

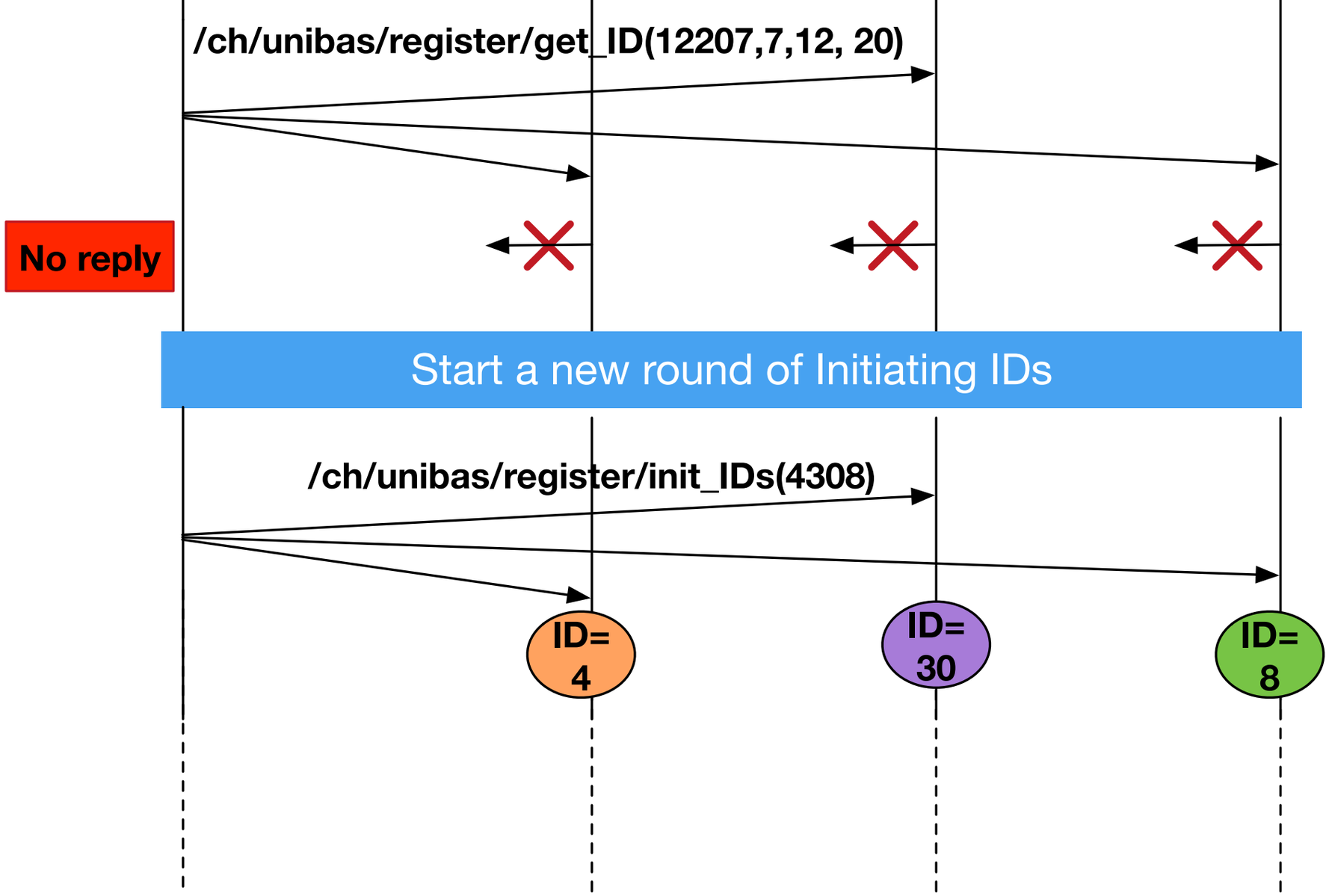
ID= 7

/ch/unibas/register/pull_info(12207, 20,L)

Info{L}

/ch/unibas/register/get_ID(12207,7,12, 20)

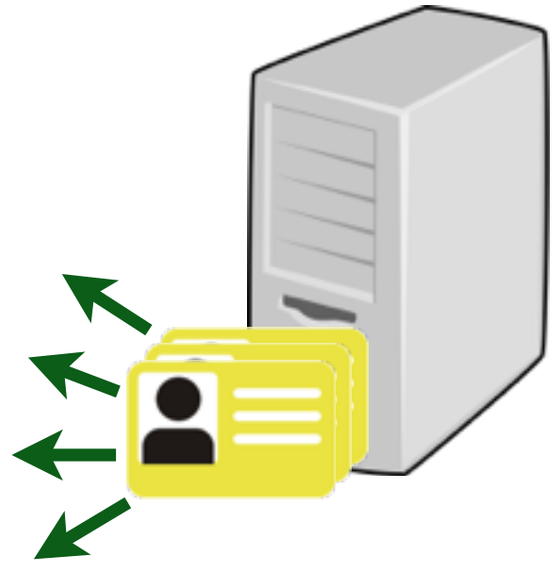
No reply


/ch/unibas/register/get_ID(12207,7,12, 20)
No reply
Start a new round of Initiating IDs
/ch/unibas/register/init_IDs(4308)
ID= 4
ID= 30
ID= 8

Discussion

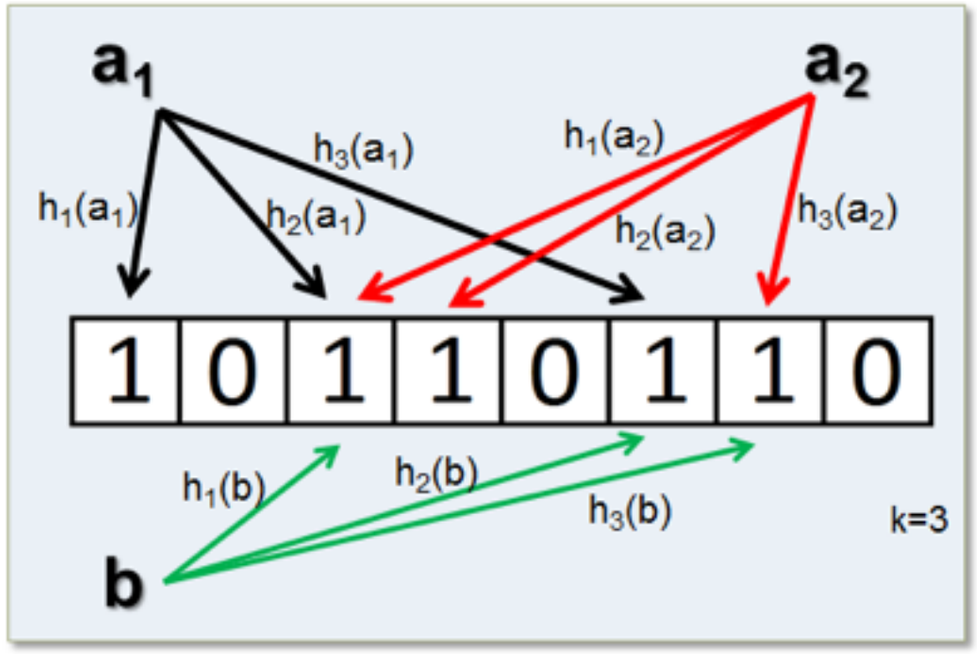
- *Replica ID possibilities*

61424	20419	86546	00517
90222	27993	04952	**20631**
50349	71146	97668	86523
85676	10005	08216	25906
02429	19761	15370	43882
90519	61988	40164	15815
20631	88967	19660	27932



- *get_ID(…) parameters*

In huge data center —> huge number of exclusions —> Bloom filter



Discussion (loose ends)

- *Failure Discovery* —> Every time, the replica replies with its new ID, it sends a list of the previous IDs

- *Push or Pull* —> NCMP is Pull architecture, could NCMP have push architecture

- Group membership management protocol (heartbeats, join and leave messages) ?

- What if the controller needs to *reset/reconfigure/ initiate* a replica ?

Summary and future work

- NCMP as a monitoring protocol (with 3 primitives) to scan all service replicas and pull run-time information

- Can benefit from improvements, e.g. forwarding strategy (to the best replica)

- Next level challenges: full management actions (quickly discovering a failed replica, spin-up another)

- Implementation on the way using CCN-lite and NFN