TOWARDS CONTROLLER-AIDED MULTIMEDIA DISSEMINATION IN NAMED DATA NETWORKING

Florian Bacher, Benjamin Rainer, Hermann Hellwagner
Multimedia Communication Group (MMC)
Institute of Information Technology (ITEC)
Alpen-Adria-Universität (AAU) Klagenfurt
{firstname.lastname}@itec.aau.at

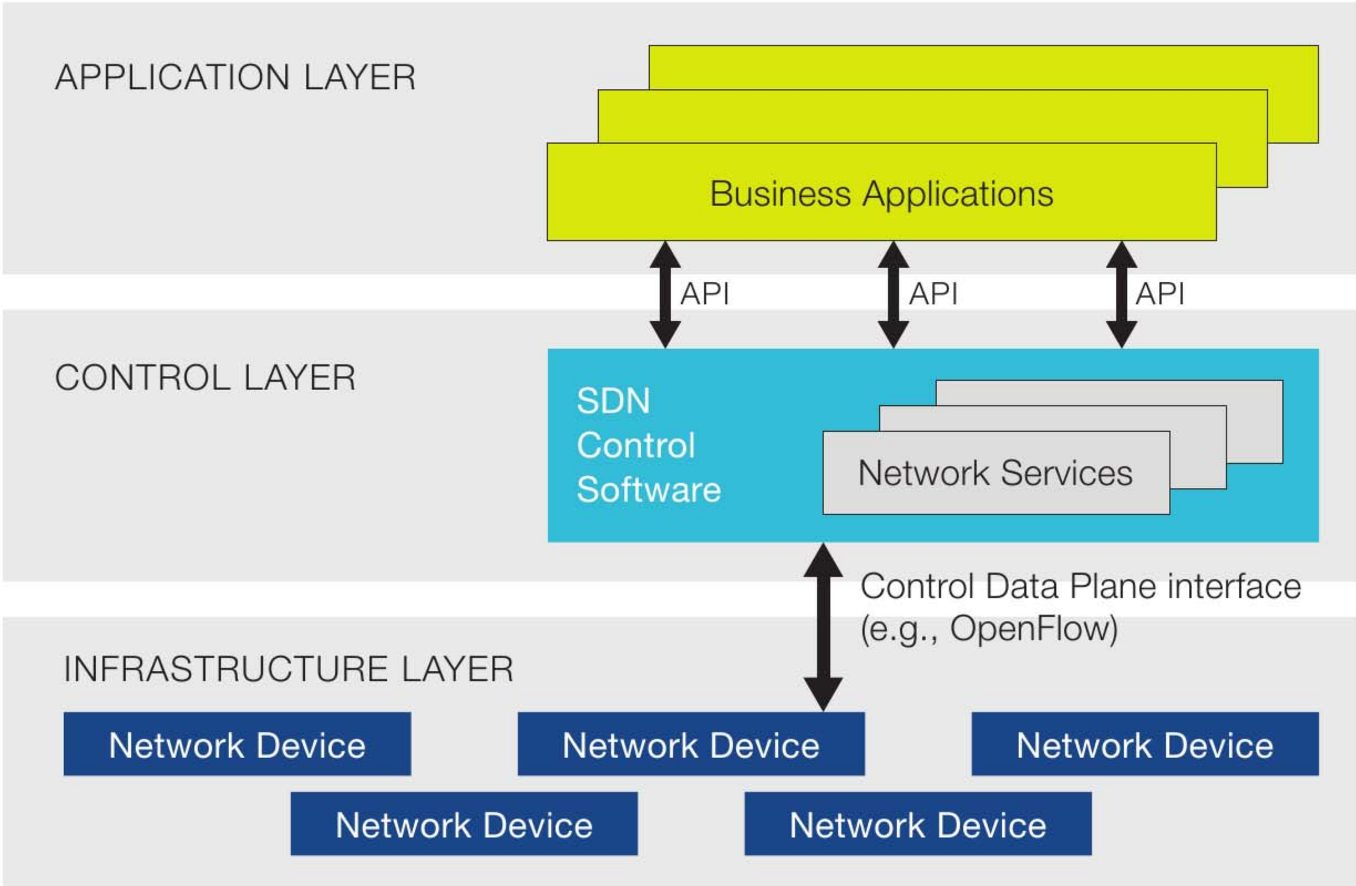
INTRODUCTION

- Internet traffic nowadays focused on content rather than the direct communication between two hosts
- Named Data Networking (NDN):
  - Clients send Interests with the name of the content to the network
  - Inherent caching of content chunks on each router
    - Routers check if content for a received interest is available and send back data
    - No need to forward Interest to content origin if cached copy of a requested chunk is available
  - Separation of routing and forwarding
    - Forwarding decisions can be done independently from routing protocols

INTRODUCTION

- One big challenge in NDN:
  - Naming of content rather than hosts leads to a large address space
- Possible solution: Software Defined Networking (SDN)
  - Separation of control and data plane
    - Control plane moved to a central controller
    - Networking elements are reduced to simple forwarding elements
    - Easy deployment of network wide policies
    - Developed independently from ICN/NDN, but aligns well to the separation of routing and forwarding in NDN

SDN ARCHITECTURE

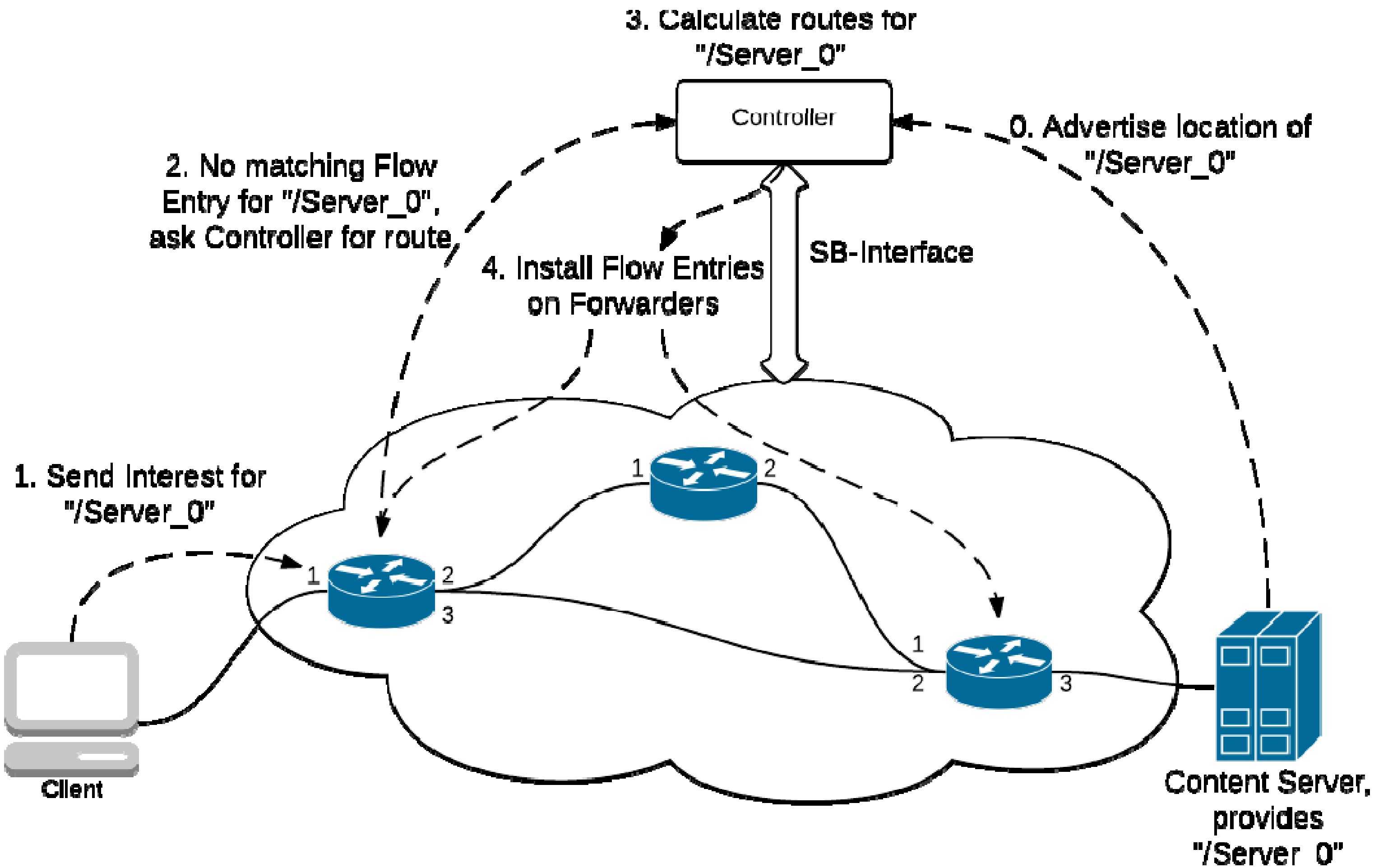


Source: www.sdxcentral.com

Opportunities of Combining SDN with NDN

- 1 Central controller which knows:
  - Network topology
  - Location of content
  - Current network load
  - Broken links
- NDN Routers
  - Request forwarding options from controller
  - Forwarding decisions are made independently (cf. decoupling of routing and forwarding)
  - Report events to the controller
- E.g., implemented by Torres et al.: Controller-based routing scheme for Named Data Network (CRoS)

PROPOSED APPROACH



LOCAL FORWARDING

- Routers receive forwarding options and their cost (i.e., hop count) from controller
- Routers keep track of face reliability (Percentage of satisfied interests)
- Hop count and reliability are taken into consideration during forwarding (cf. Algorithm 1 in the paper)
  - Face reliability has to be above a predefined threshold to be dategorized as reliable
  - Cheaper faces (low hop count) are preferred
  - If no reliable face is available, a random face is selected → check if previously unreliable faces recovered in the meantime
  - Faces must have bandwidth available (token bucket filter)
  - If no face with available bandwidth has been found → send NACK

PROACTIVE CACHING I

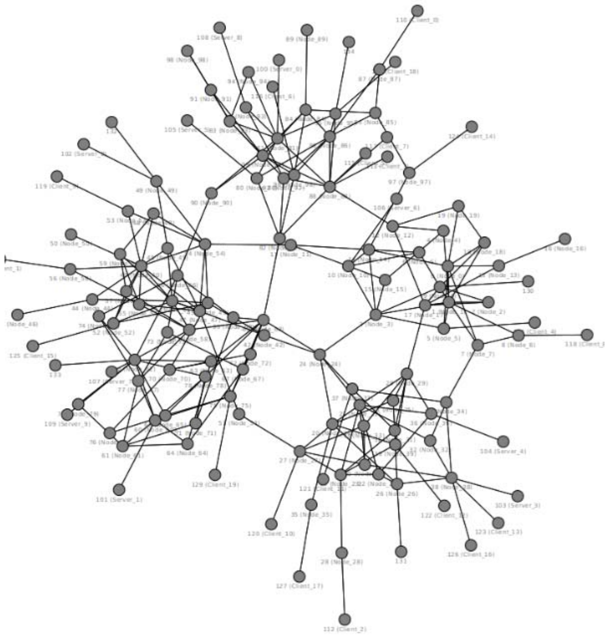
- Another major advantage of SDN:
  - Controllers can communicate with an application layer component via a northbound (NB) API
  - Application layer component can control the network behavior
  - Enables development of application-aware networks
    - Different applications require different options in terms of routing and/or caching
  - Our example use case:
    - Disseminating Zipf-distributed multimedia content (i.e., major part of traffic is caused by a small number of popular contents) over the network

PROACTIVE CACHING II

- Application layer component (AL) knows the popularity values of several contents
  - Popularity values can be different in each autonomous system (AS)
  - AL can instruct dedicated nodes to proactively cache content
  - Assumption: Popularity values are known by AL
  - Future work: continuously monitor traffic and try to predict traffic patterns/popularity distribution

EVALUATION

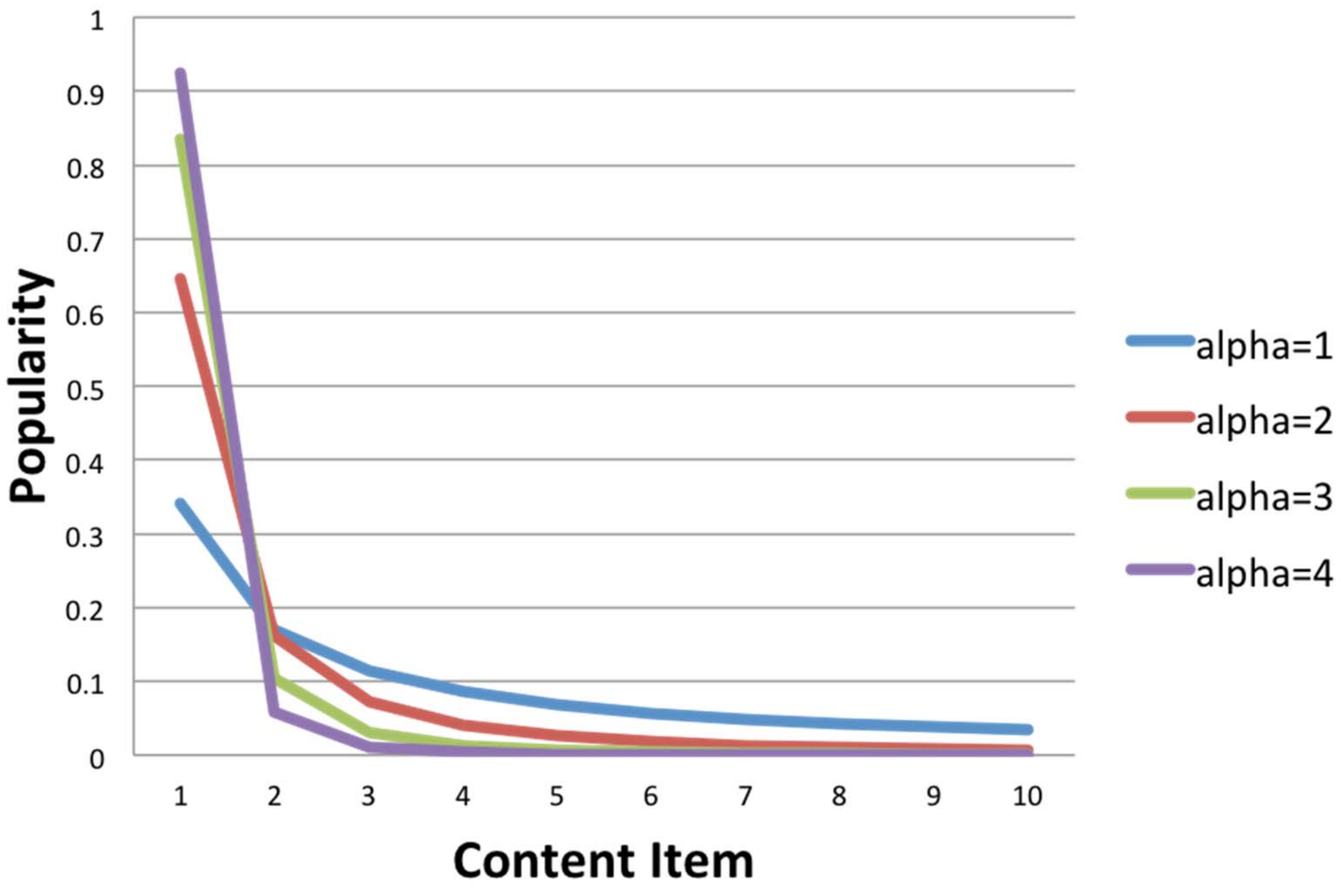
- Implementation in ndnSIM 1.0
- Topologies generated with BRITE network generator
  - 5 autonomous systems (AS) with 20 nodes each
  - Randomly distributed 10 content servers and 20 clients
  - One dedicated node for prefetching content per AS
- Compared strategies:
  - BestRoute
  - SDN without proactive caching
  - SDN with proactive caching
- 30 scenarios/simulation runs per strategy



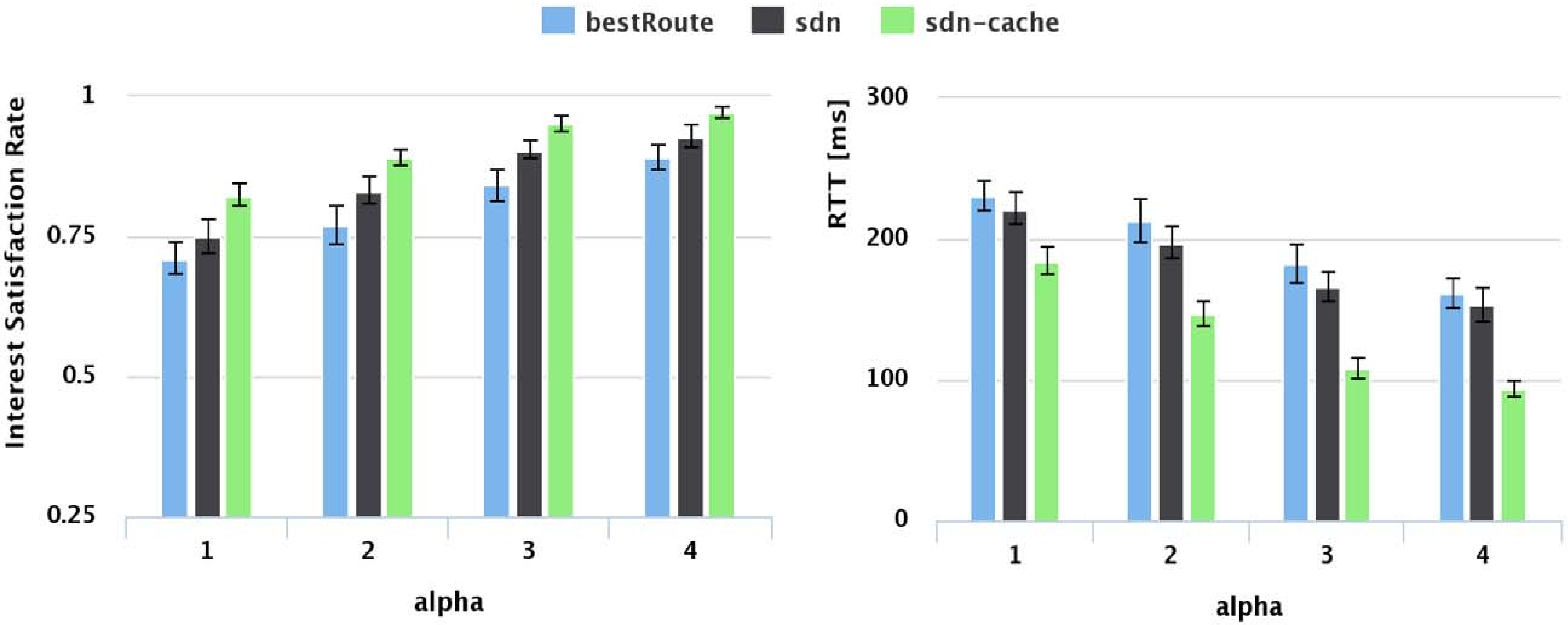
EVALUATION II

- Simulation runs were divided into four periods
  - 80 seconds per period
    - 40 seconds of high network load: clients request files of 5MB with constant rate of 30 Interests per second (~1 Mbps)
    - 40 seconds of low network load: Proactive caching during this phase
- Network bandwidth was configured such that **congestion is likely to occur** during high load phases (2-4 Mbps between AS, 1-2 Mbps within each AS)
- Evaluated for different values of parameter *alpha* of the Zipf-distribution (1,2,3,4)

EVALUATION II: INFLUENCE OF ALPHA



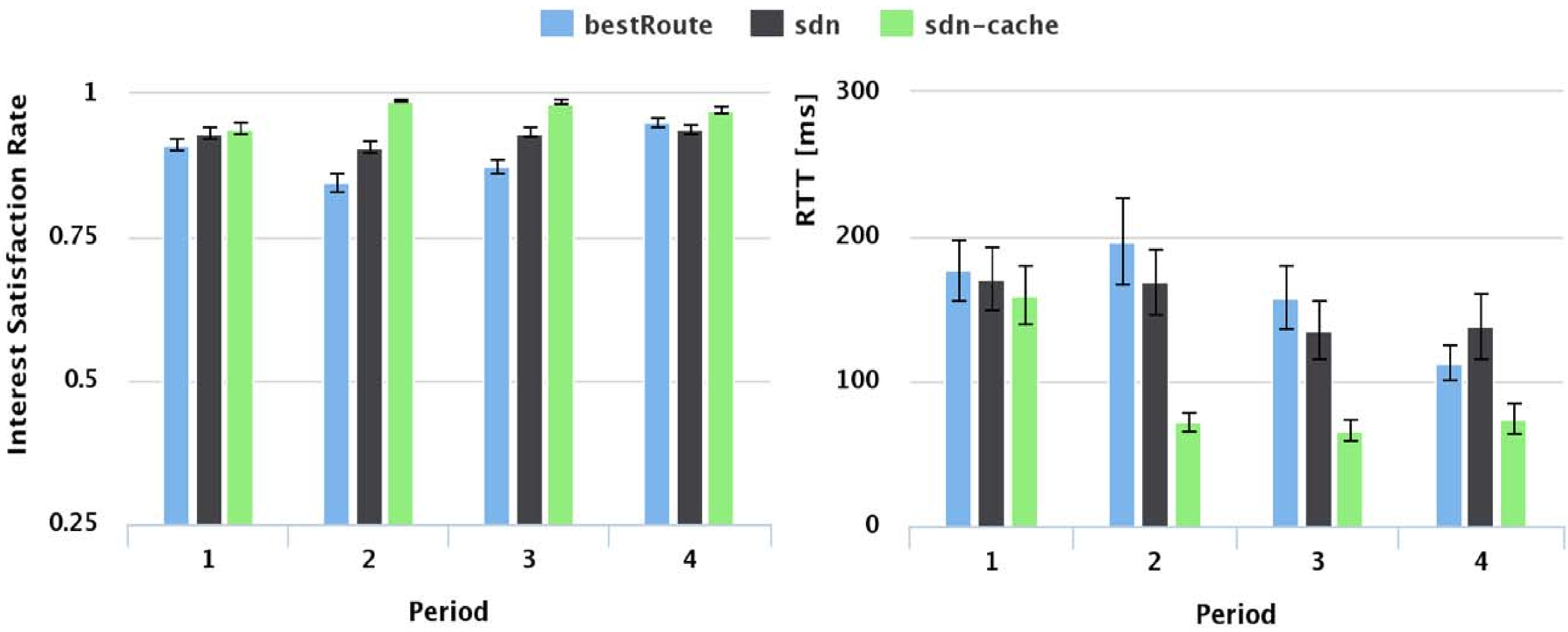
Results: Overall Interest Satisfaction Rate and Interest/Data RTT with 95% CI



Overall Average Interest Satisfaction Rate

Overall Average RTT

Results: Satisfaction Rate and RTT per Period, alpha=4



Interest Satisfaction Rate per Period

RTT per Period

CONCLUSIONS AND FUTURE WORK

- Holistic view of the network enables quick reaction to events such as link congestion
- Promising results for controller-aided forwarding strategy
- Significant improvement when combined with proactive caching
  - Popularity and traffic predictions assumed to be known at the application layer in our scenarios
  - Further investigation of possibilities and limitations of predicting network traffic and content popularity necessary
- Investigation of controller comm. overhead necessary
- Implementation with OpenFlow

THANK YOU FOR YOUR ATTENTION!

Questions?