On Bloom Filter-based Routing in Information-Centric Networks

July 24, 2017

Ali Marandi
Communication and Distributed Systems
Institute of Computer Science
University of Bern, Switzerland
marandi@inf.unibe.ch

b UNIVERSITÄT BERN

UNIVERSITÄT BERN

Summary of the talk

- **Gist of the talk:** A comparative performance analysis of BFR and COBRA routing approaches
  - Overview of BFR
  - Overview of BFR performance analysis
  - Description of COBRA
  - Comparative performance analysis for BFR and COBRA routing schemes

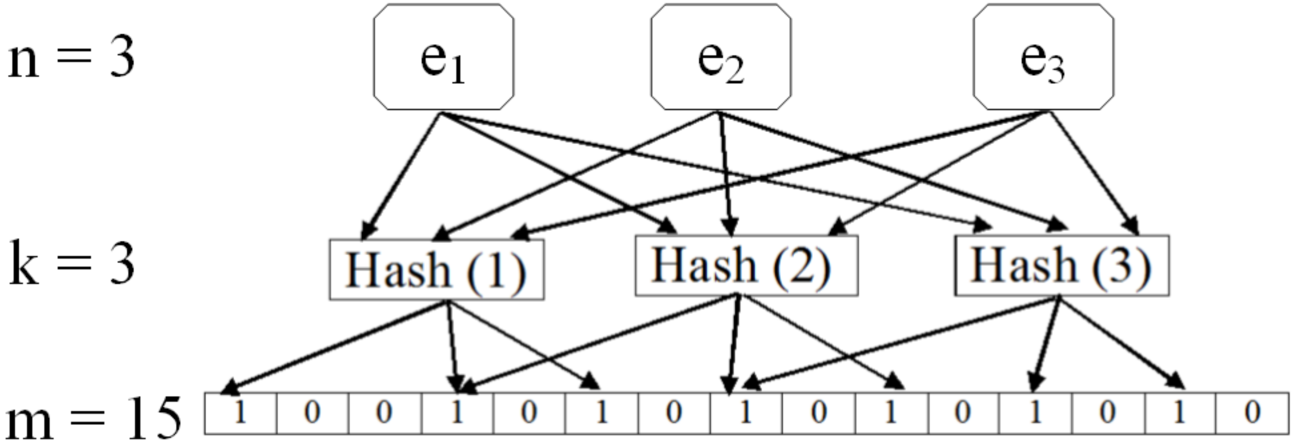
- **Starting subject:** Different types of Bloom Filters (BFs)
- **Ending content:** Future work

Different types of BFs

u^b

UNIVERSITÄT BERN

BF (let us call it *classical* BF)



- **Benefits**: representing large sets in a compact way, allowing fast membership queries independent of set size
- **Benefit in networking**: reduced overhead of transferring or storing information
- False negative errors are impossible
- Dealing with false positive errors is the only price to pay
- It is not allowed to delete elements from the classical BF

UNIVERSITÄT BERN

Theoretical tradeoffs

- ***m***: BF length (bits)
- ***n***: number of inserted elements
- ***k**:* number of hash functions

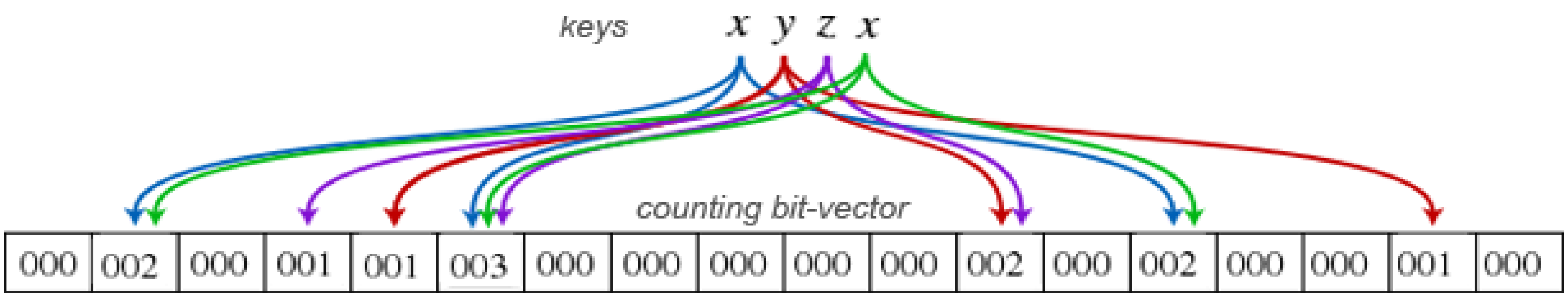
$$k = \frac{m.\ln 2}{n}$$

- ***p***: false positive probability

$$m = \frac{-n \ln p}{(\ln 2)^2}$$

Counting Bloom Filter (CBF)

- **Main feature:** deletion operation is allowed
- **How?** Using a table of *counters* rather than using a *bit* table
- An example of a CBF with four hash functions and 12-bit counters

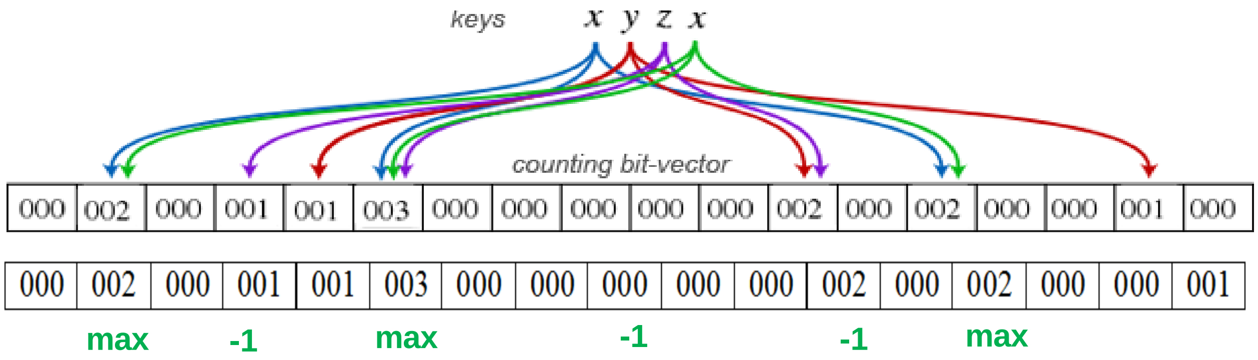


- Insertion operation consists on incrementing the associated counters and deletion operation consists on decrementing the associated counters
- **Problem1:** frequent insertions might lead the counter to overflow (false positive errors)
- **Problem2:** frequent deletions might lead the counter to underflow (false negative errors)

UNIVERSITÄT BERN

Stable Bloom Filter (SBF)

- Is a CBF
- Insertion operation consists on setting the associated counters to their maximum value as well as randomly decrementing $l<N$ counters (assuming N the total number of counters)



- Deletion operation has the same mechanism, like CBFs
- **Benefit:** storing the most fresh elements over time and removing stale elements gradually

Overview of BFR

UNIVERSITÄT BERN

A summary of features for BFR scheme

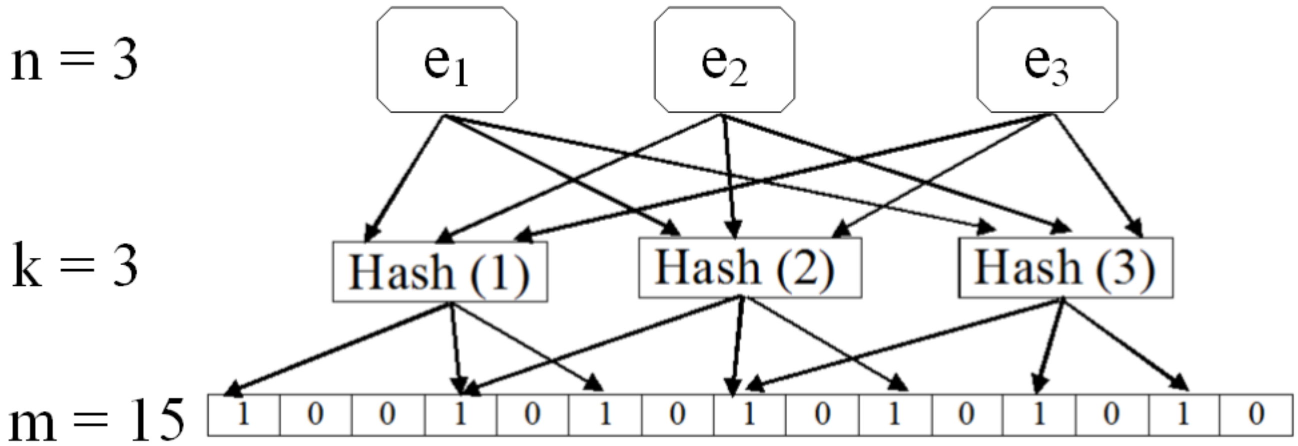
- Intra-domain
- Topology oblivious
- Fully content-oriented
- Does not adopt any IP-based approach
- Works based on content advertisements
- Uses classical BFs for content advertisement
- Reasonable storage and signaling overhead for content advertisements
- BFR benefits from multi-path content discovery by using the *multicast* forwarding strategy

BFR operation

- Representation of content objects using BFs
- BF-based content advertisement
- Content retrieval and FIB population
- Handling of
  - False positive errors
  - Topology changes
  - Content migration

Representation of content objects with BF

$$\begin{cases} e_1 = /unibe.ch/, \\ e_2 = /unibe.ch/images/, \\ e_3 = /unibe.ch/images/fileName1 \end{cases}$$

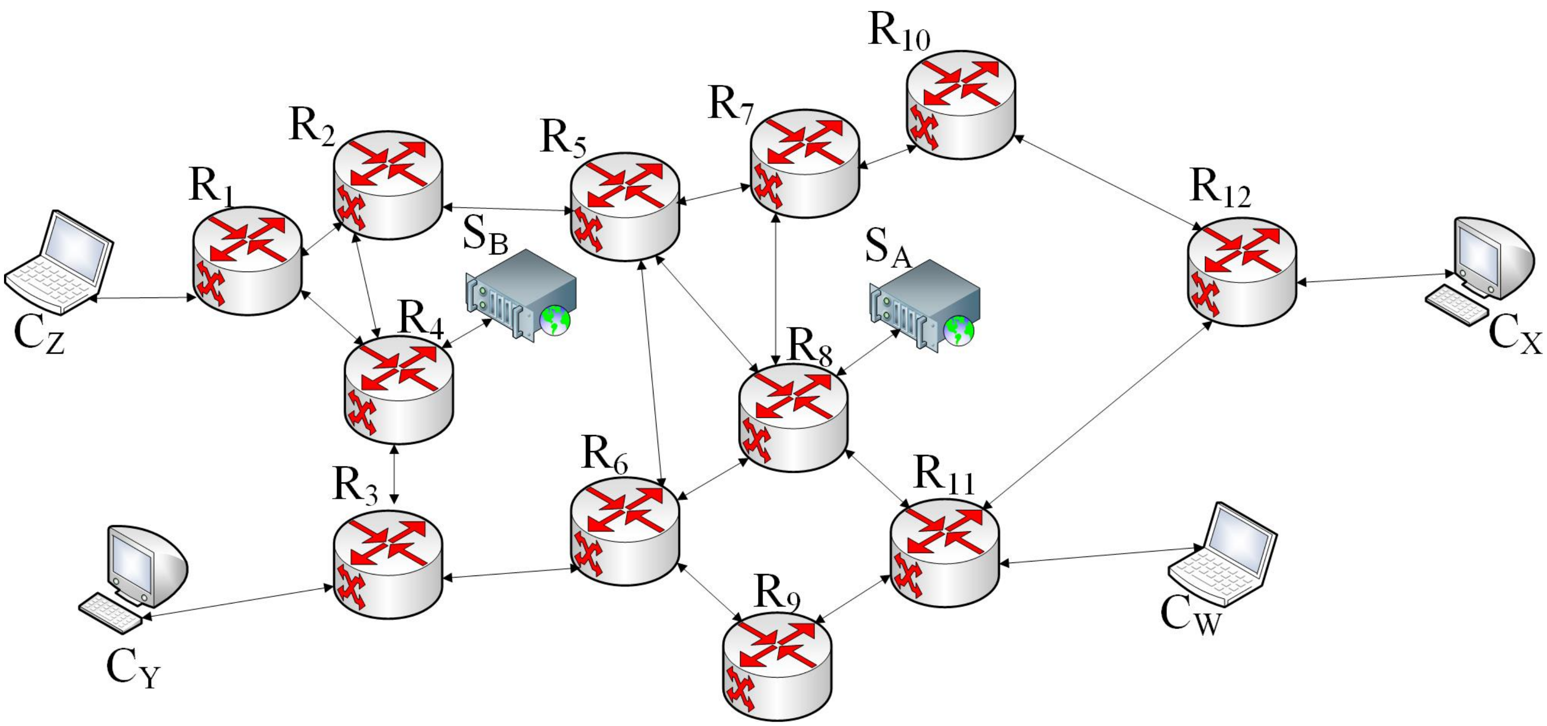


UNIVERSITÄT BERN

BF-based content advertisement

PIT TABLE OF C_Z

$/ContentAdvertisement/A$
$/ContentAdvertisement/B$
$/unibe.ch/images/fileName1/01$



UNIVERSITÄT BERN

Impact of false positive errors on BFR

- What if a false positive error occurs ?
  - BFR benefits from multi-path content discovery, thus the Interest will be routed towards both the correct and wrong origin servers
  - But the Interest will be anyway satisfied, because it has been routed towards the correct origin server
  - The Interest *might reach* a wrong origin server
- All the Interests will be satisfied in presence of false positive errors

UNIVERSITÄT BERN

Robustness to topology changes

- Resiliency to link failures
  - Avoiding all the Interests from going through the failed link
    - Removal of the face associated with the failed link from all the FIB entries
    - Removal of the face associated with the failed link from all the in-records of all the stored CAIs

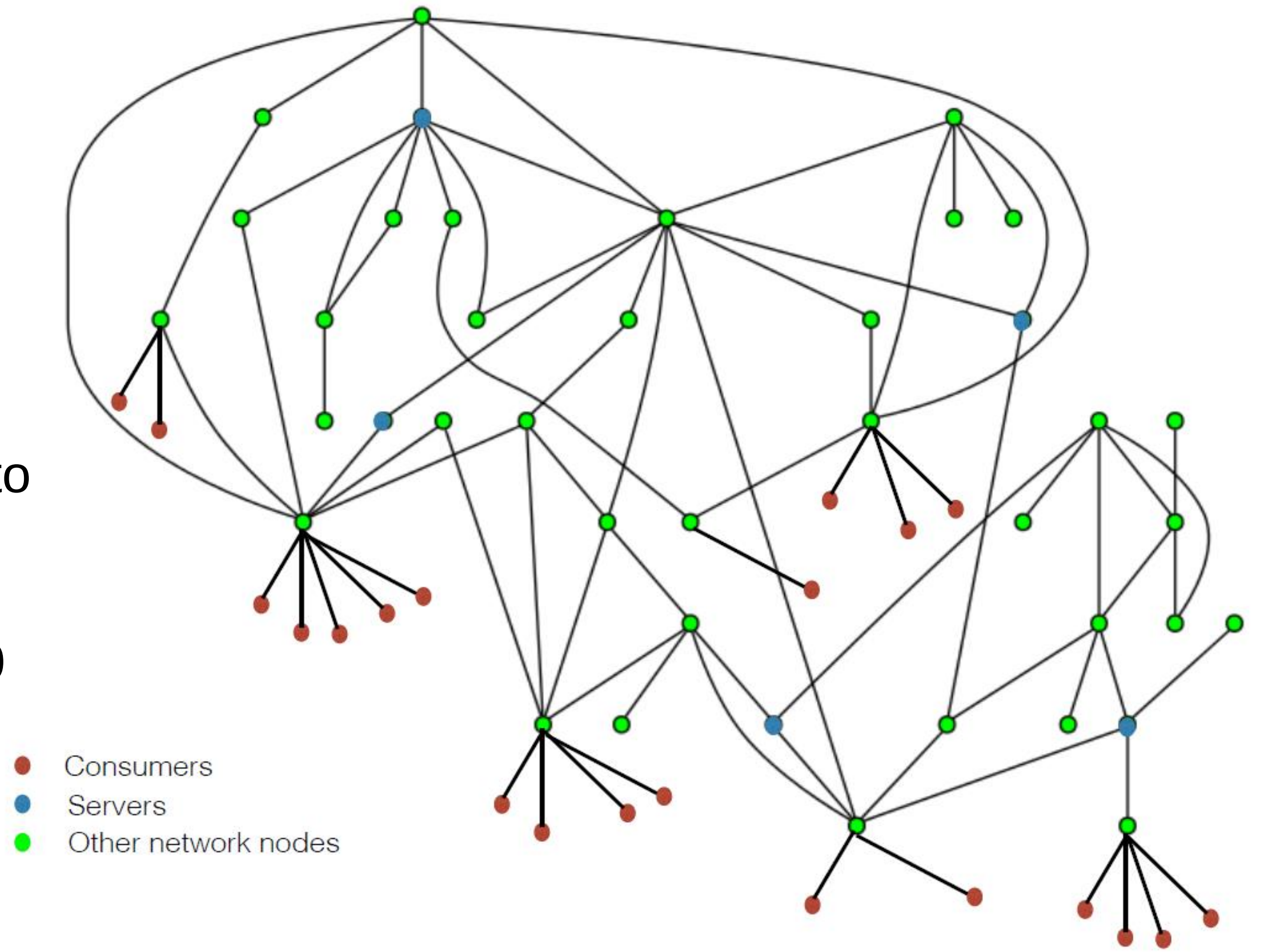
- Adaptation to link recoveries
  - Enforcing all the Interests to pass through the recovered link
    - Adding the face associated with the recovered link to all the FIB entries
    - Adding the face associated with the recovered link to all the in-records of the stored CAIs

BFR performance analysis

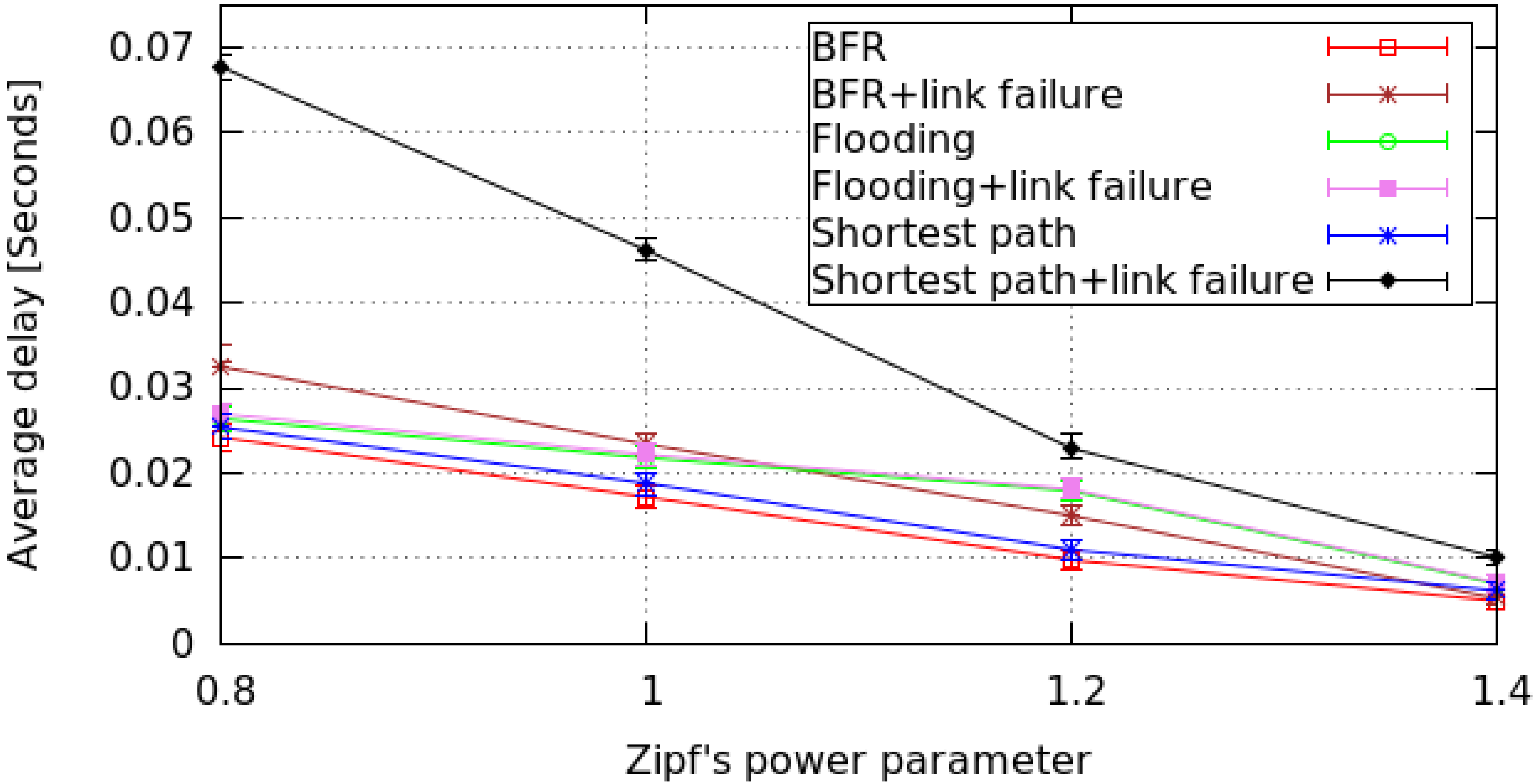
UNIVERSITÄT BERN

Simulation settings

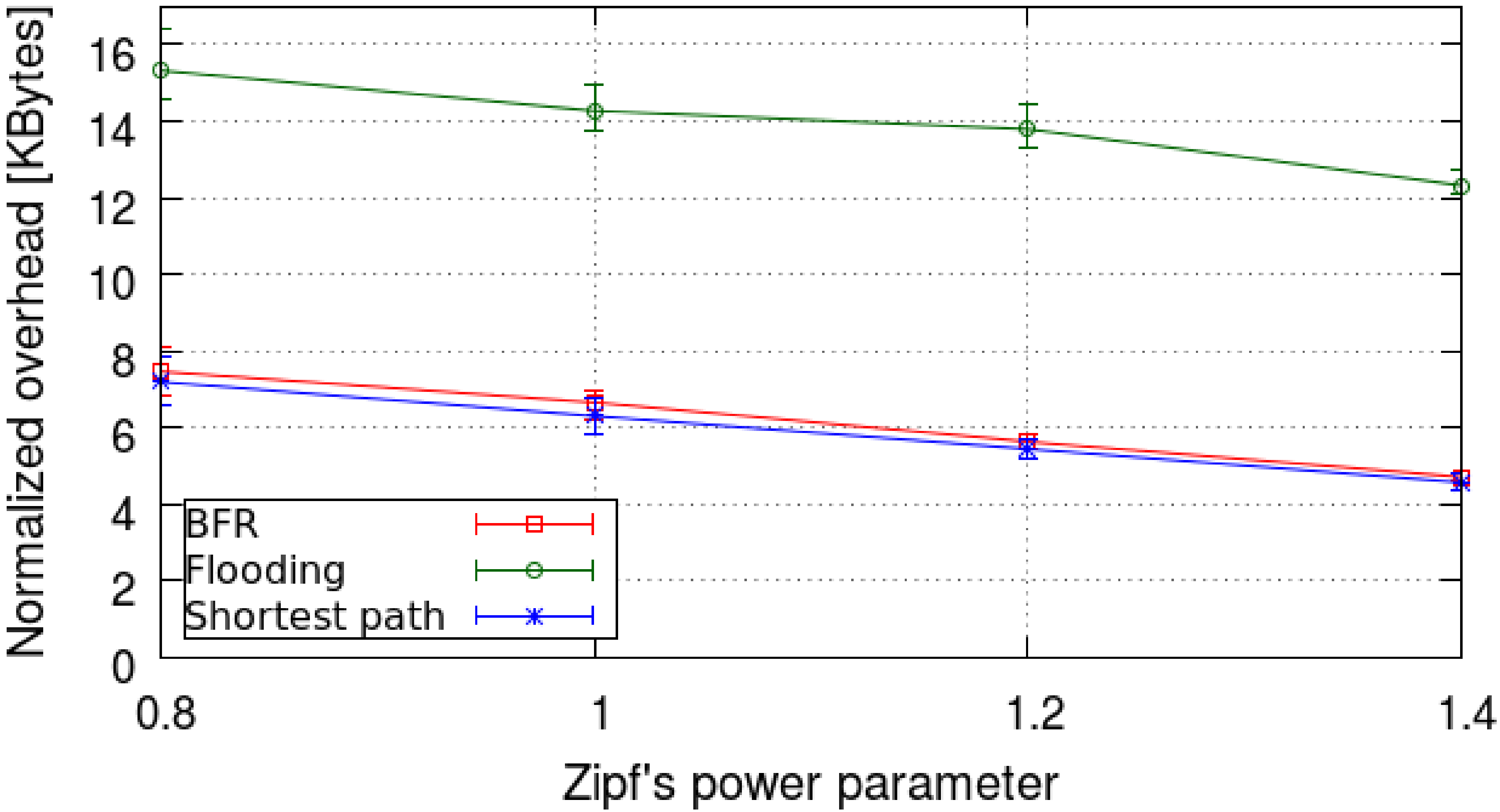
- 101 nodes
  - 39 routers
  - 56 consumers
  - Five servers
- URL catalogue
  - 1000 URLs
  - Each divided to 100 segments
- CS capacity = 100
- Zipf-like popularity
- Simulation time of 28 hours



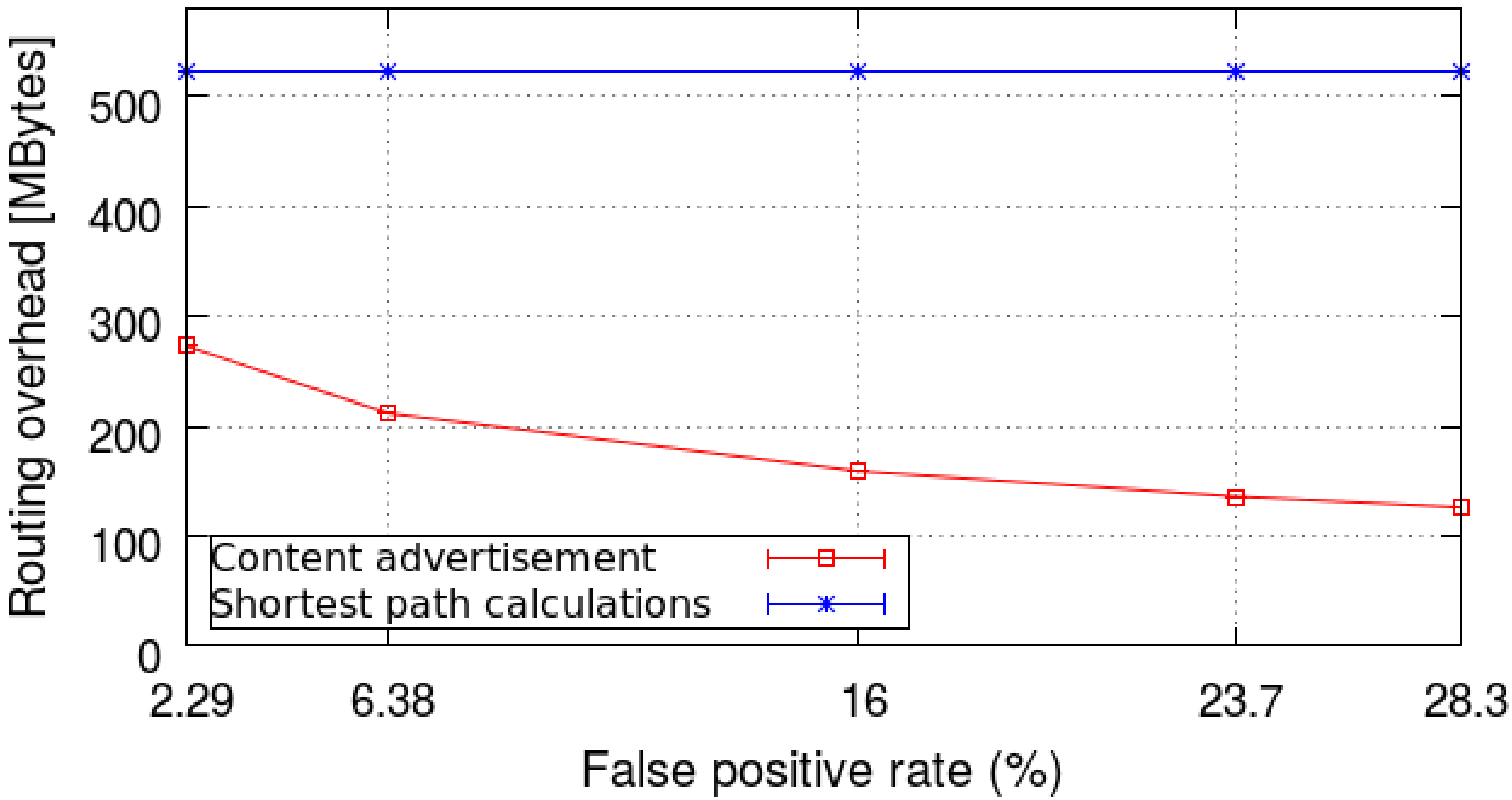
Results for average round-trip delay



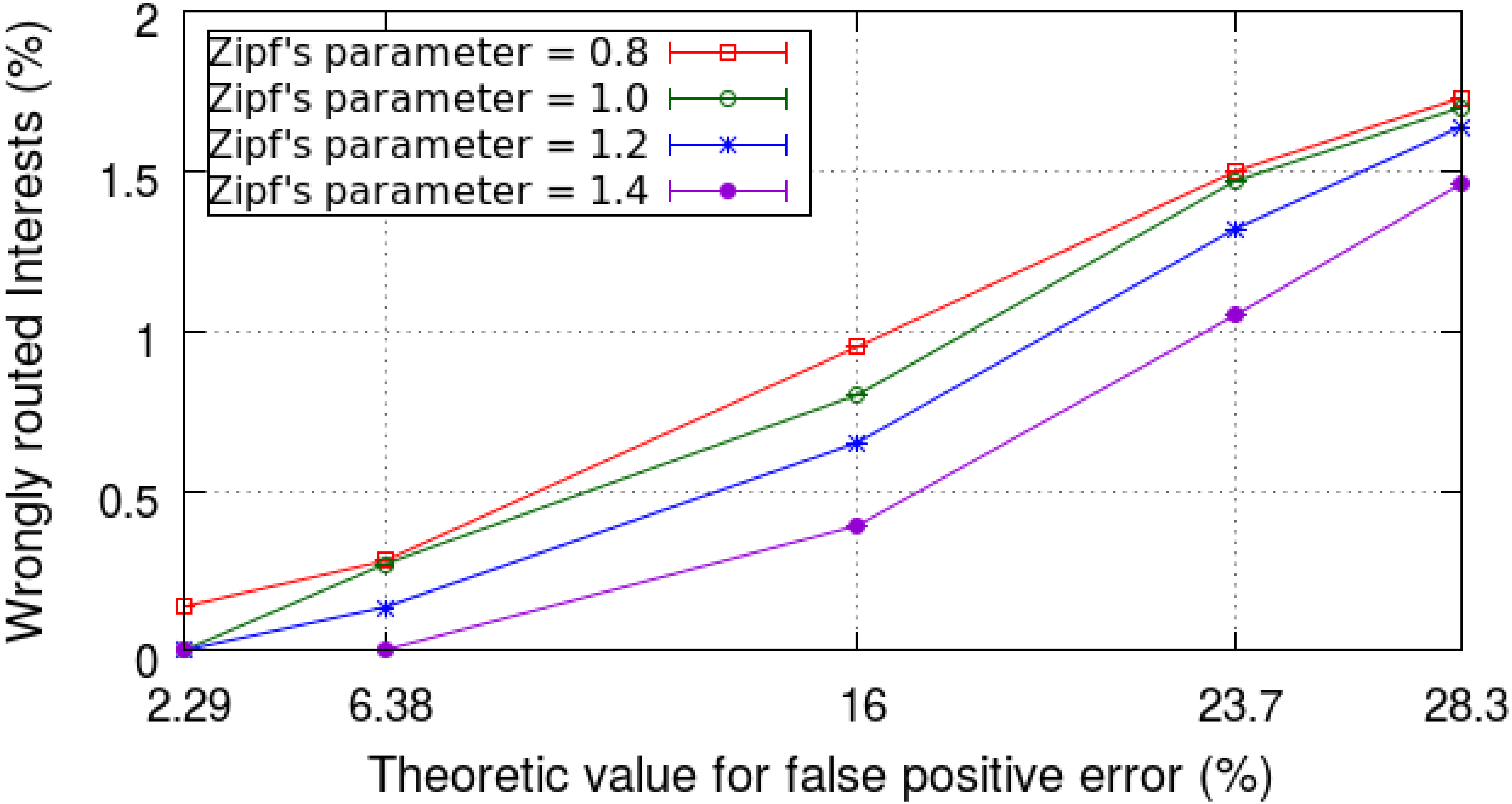
Results for communication overhead



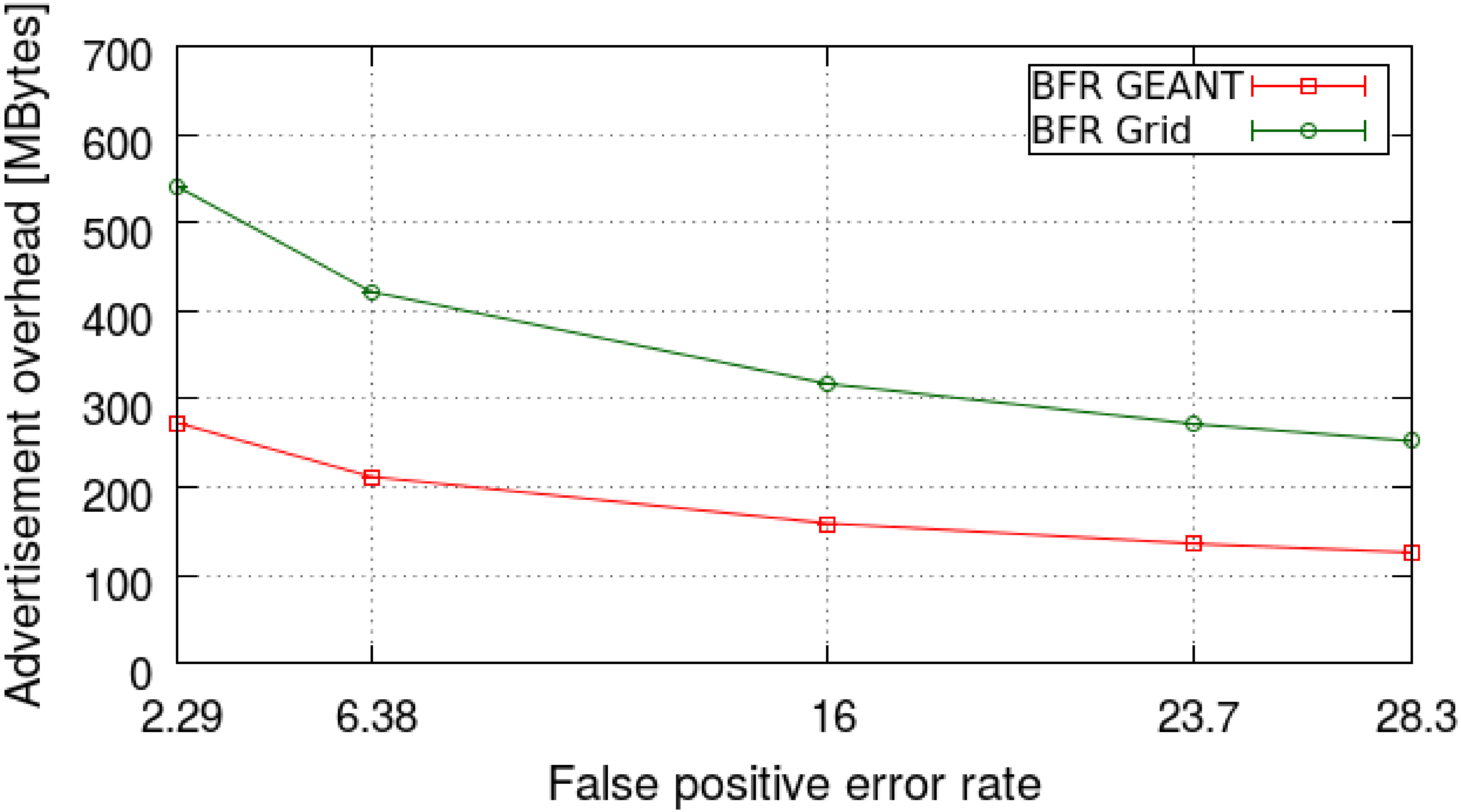
Results for routing overhead



Impact of false positive error on BFR

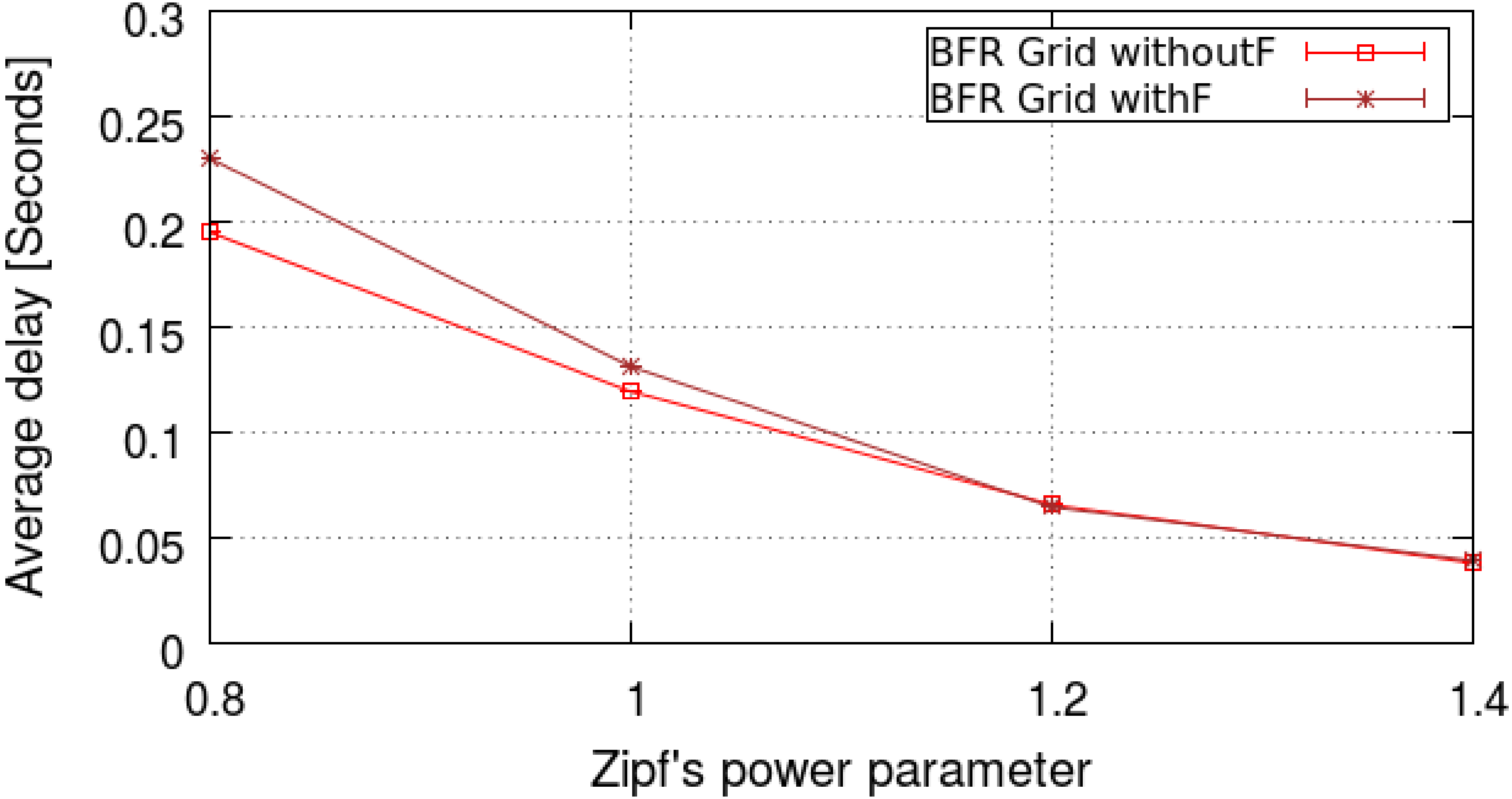


Content advertisement overhead



UNIVERSITÄT BERN

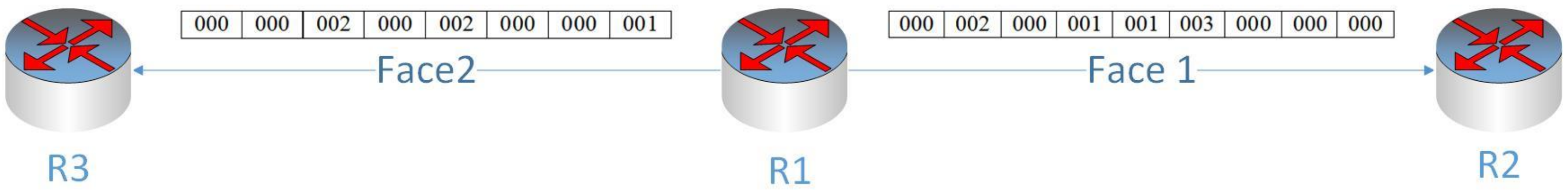
Average round-trip delay



UNIVERSITÄT BERN

COBRA routing scheme in a nutshell

- Routes Interests according to *breadcrumbs*
- Uses Stable SBFs for storing breadcrumbs
  - Each node assigns one SBF per face for this purpose

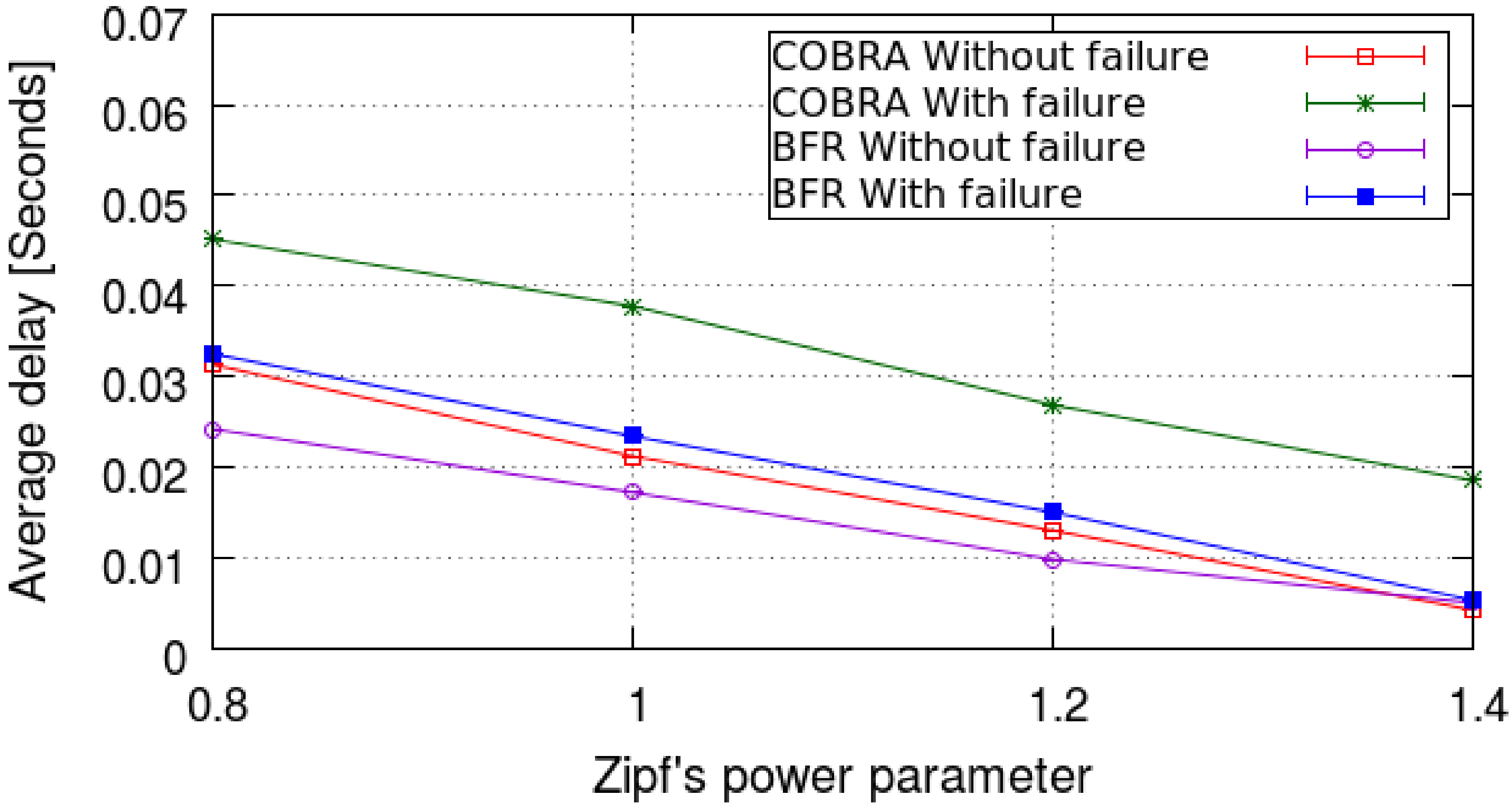


- Before learning the routes, simply floods the Interests
  - Higher communication overhead for Interest diffusion
- Does not benefit from multi-path content discovery (uses Best-Route strategy)
- Upon detecting a link failure, resets the associated SBF to avoid forwarding Interests over the failed link
- Upon detecting a link recovery, sets the associated SBF to encourage Interests to go through the recently recovered link

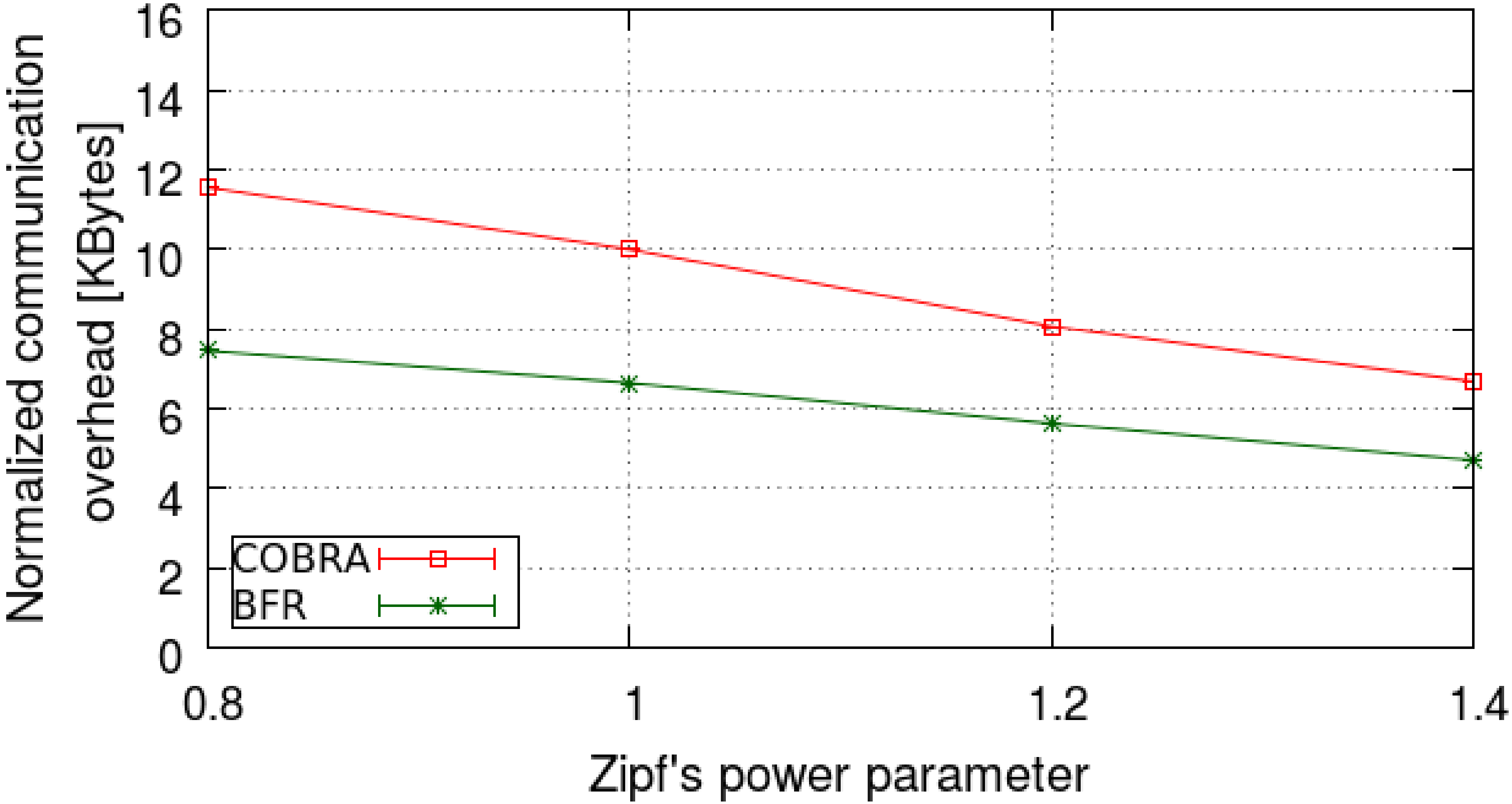
UNIVERSITÄT BERN

Comparative performance analysis of BFR and COBRA routing approaches

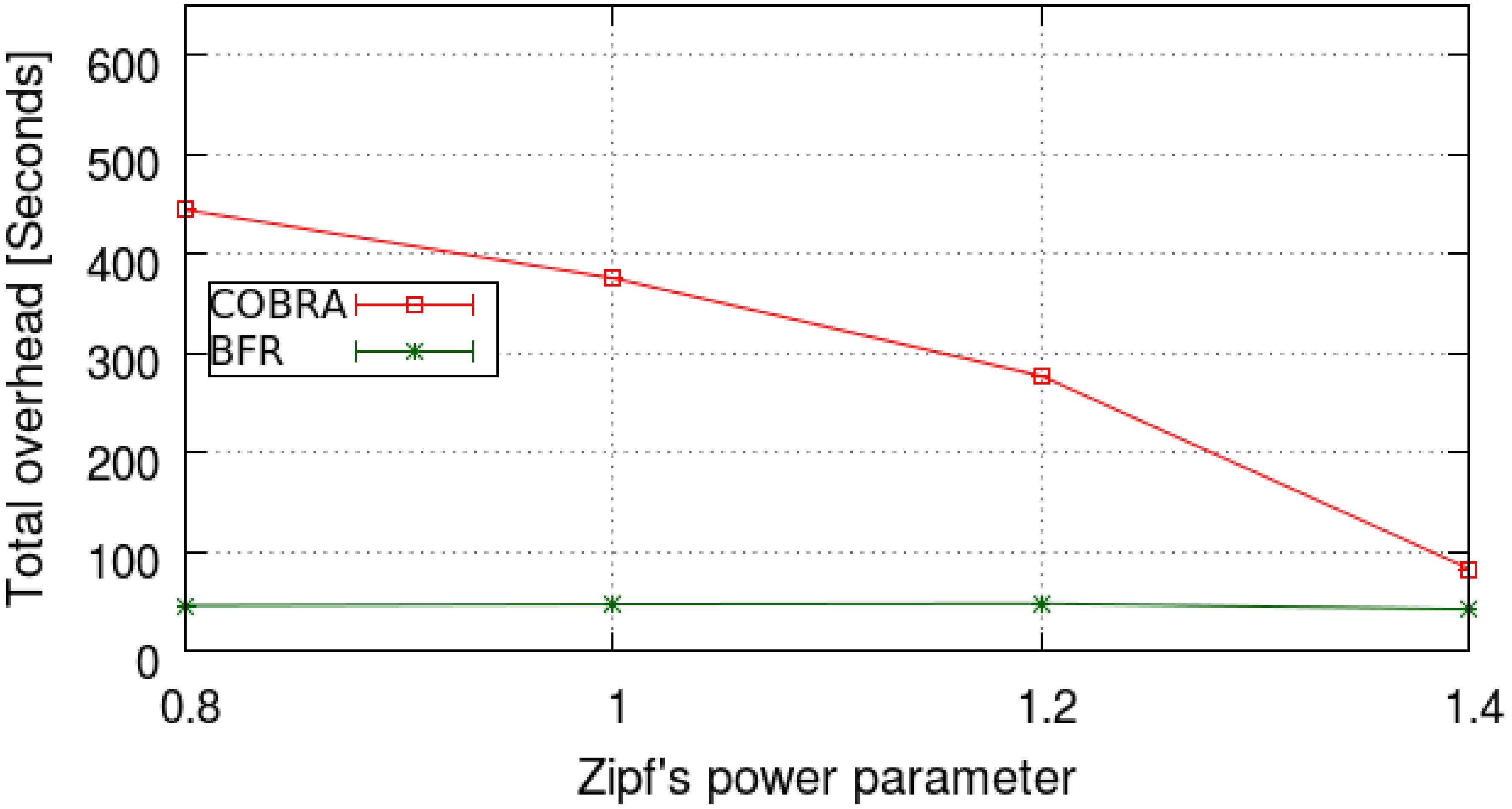
Average round-trip delay comparison



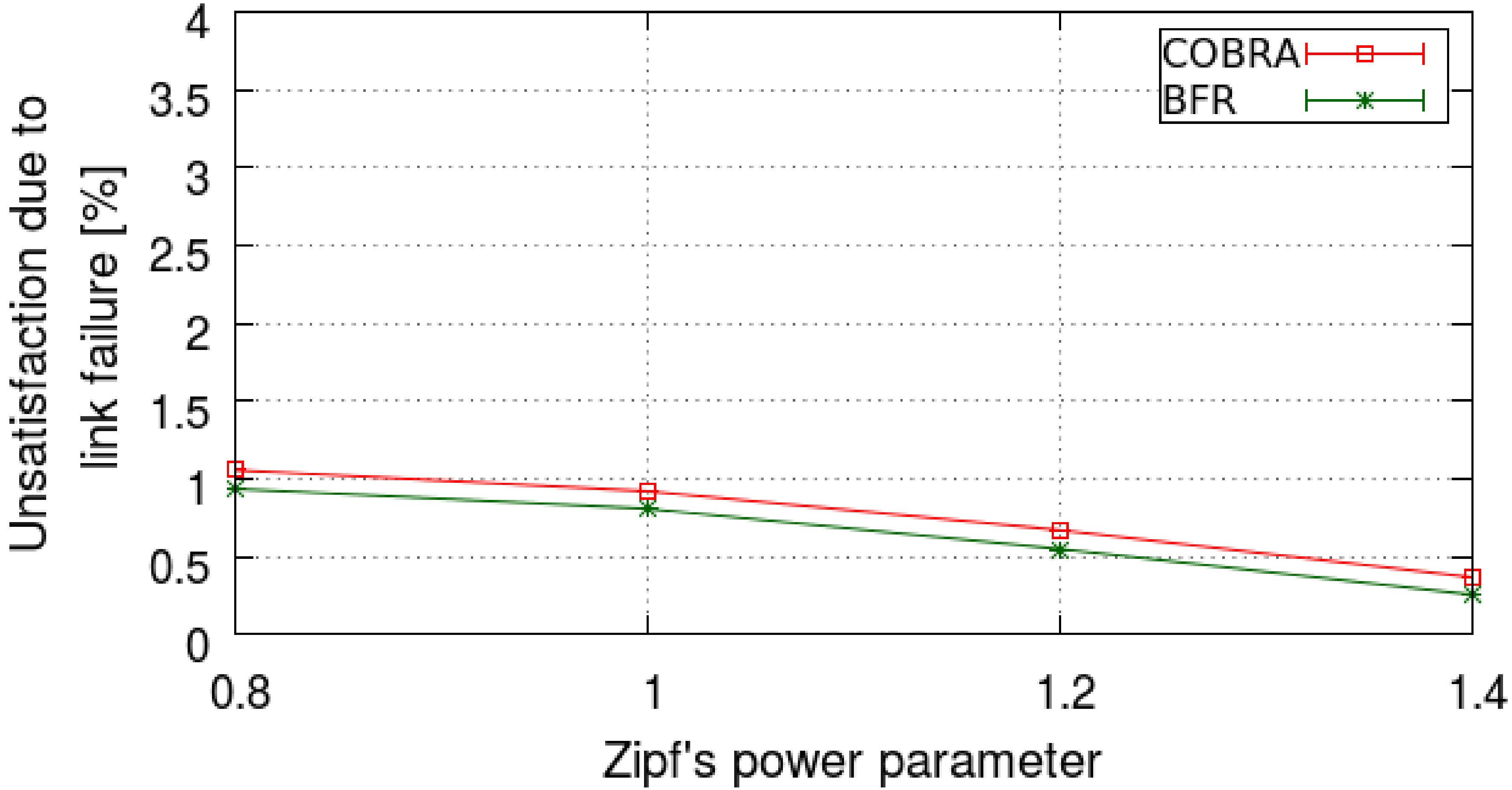
Normalized communication overhead



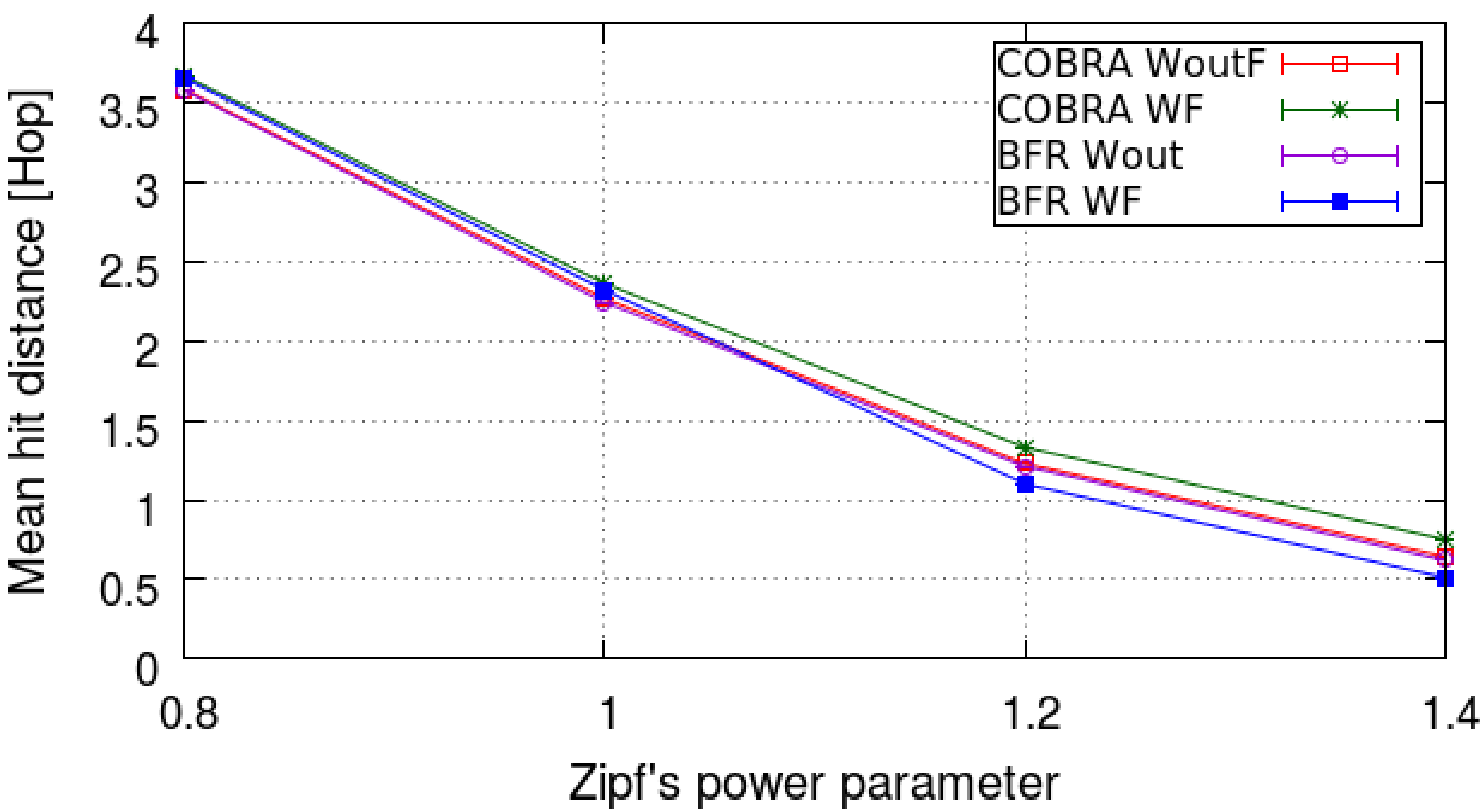
Total communication overhead for Interests



Dissatisfaction due to link failure



Mean hit distance comparison



UNIVERSITÄT BERN

To do

- BF aggregation
- Enlarging the content universe size
- Testing with other topologies

Thank you !
Questions ?