Cooperative Caching in Space Information Networks

dr. Anders Fongen, May 2022
Norwegian Defence University College, Cyber Defence Academy, Lillehammer
email: anders@fongen.no

INTERNET2022, Mestre, Italy

Presenter’s bio

Anders Fongen

- Associate Professor, Norwegian Defence University College
- Field of research: Distributed Systems, Networking security
- PhD in Distributed Systems, Univ. of Sunderland, UK, 2004
- Career history
  - 5 years in military engineering education
  - 10 years research in defence research (Chief Scientist)
  - 8 years in civilian college (Associate professor)
  - 11 years in oil industry
  - 6 years in electronics industry

Introduction

- The evolution of satellite communication?
  - Application Services (“Cloud Computing in Space”)
  - Higher System Complexity (larger state space)
- What are the advantages?
  - Very Low Latency (as low as 3 ms)
  - Global coverage
- Interesting property of a Low Earth Orbit (LEO) system
  - Long idle periods (due to inhabited surface) mixed with traffic peaks
- Viewed as a problem of *Distributed Computing*
  - *having a set of distinct properties*

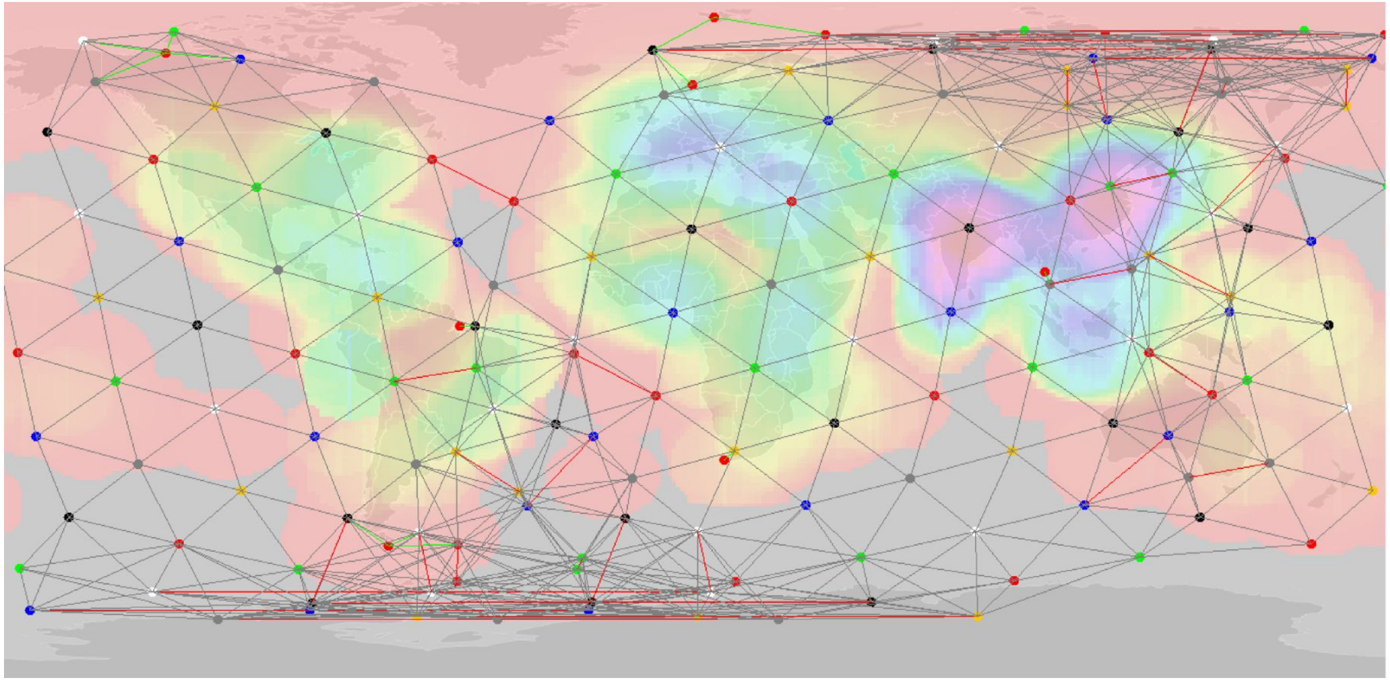
What is a SIN (Space Information Network)?

- A collection of communicating LEO satellites
- Able to serve terrestrial/airborne client
  - Communication services (e.g., IP transport, VoIP, Publish-Subscribe comm.)
  - Discovery Services (DNS, Service Brokering...)
  - Storage Services (Content Distribution Network, caching, session states)
  - Application Services (Collaborating editing, Situational awareness ...)
- Resource constrained / disadvantaged
- Predictable workload and link availability
- “Mobile” system: Stationary clients, mobile infrastructure
- Rapid hand-over of client connection and *client state*

Importance of Discovery Services in a SIN

- Performance of Discovery Services important to overall performance
  - e.g., a slow DNS service can hamper a well-tuned Web service
- Discovery Services will often employ **caching** to a large extent
- Caches kept in satellites have different properties
  - Ground clients constantly see “new” satellites (handovers etc.)
  - Caches are **trained** by clients all over the planet
  - Caches serve a huge population (storage restraints)
    - over time during orbit
    - at any moment (inside footprint)
- Solution: Distribute and coordinate caches between satellites

Population “heat map” from satellite footprint



Caches serve a *Scale Free Distribution* of queries

SFD predicts that a relative **frequency** of a (query) term is inversely proportional to its **rank**

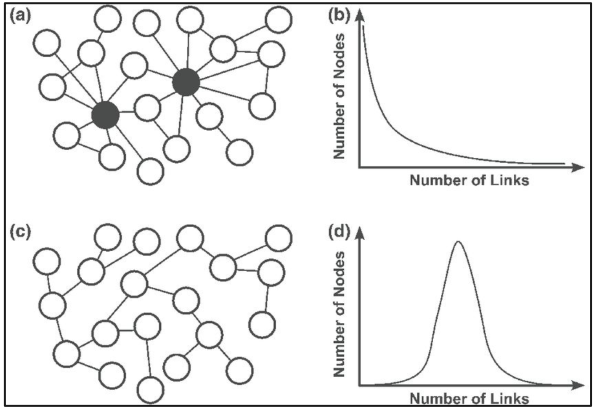
The SFD predicts that the observed frequency of a query term is inversely proportional to its *rank* r. The relative frequency f of the term t with rank r is expressed as

$$f(t_r) = \frac{a}{r} \tag{1}$$

where the value of a is determined so that

$$\sum_{r=1}^{v} \frac{a}{r} = 1 \tag{2}$$

and v indicates the size of the vocabulary in use.

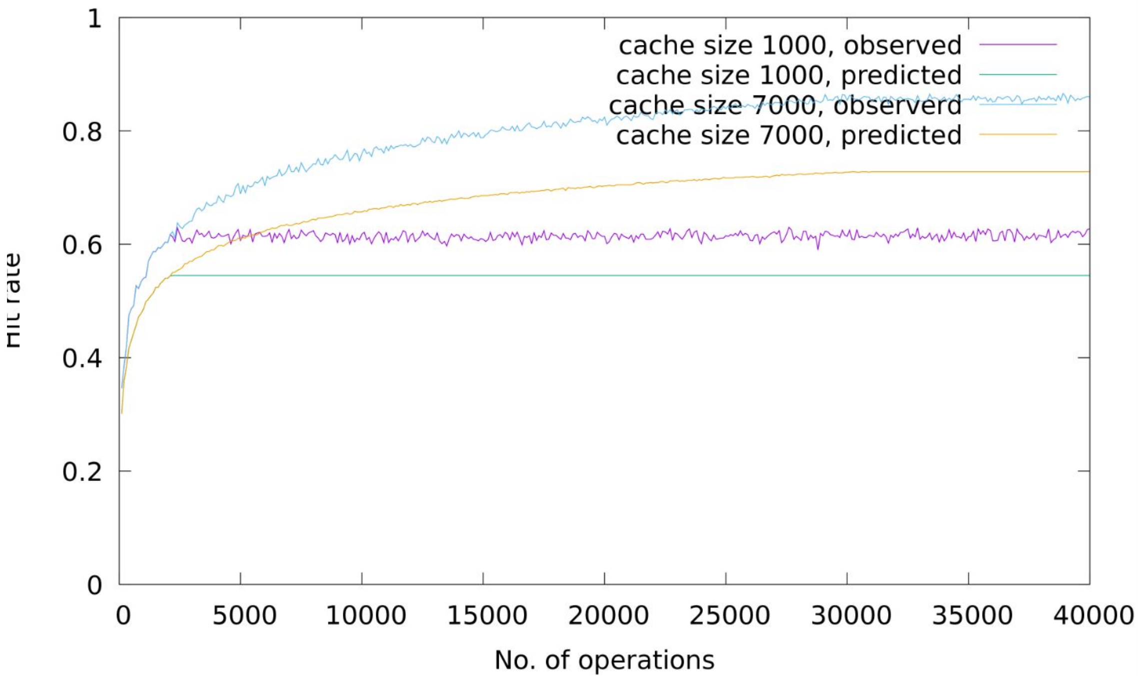


SFD causes caches to train faster

Size of vocabulary (v)=**20000**

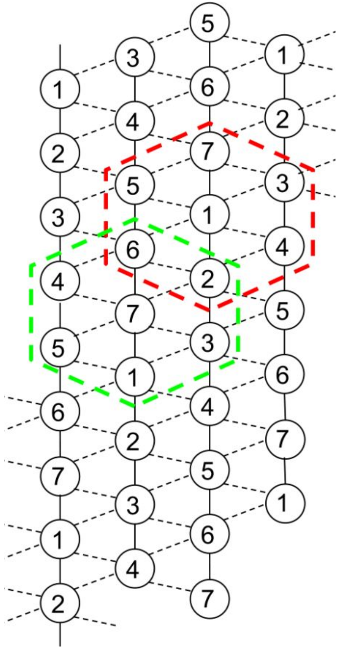
Sum frequency for the **105** highest ranked terms=**0.5**

Last-recently-used (LRU) replacement during entry insertion only **marginally** improves the hit rate



Caches in satellites constantly change audience

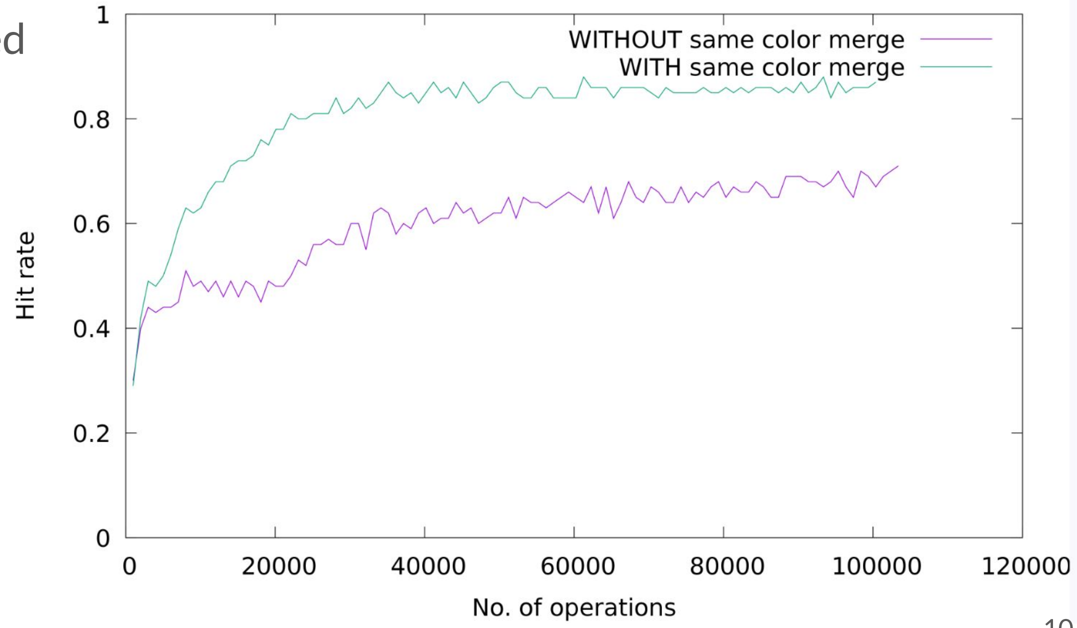
- Satellites in orbit serves the entire planet during one day
- Caches are trained through a diversity of client request
  - language, culture, time of day
- Total audience is immense
  - need to aggregate storage from many satellites
  - e.g., from those who fly in **formation**
- Cooperative caches from 7 satellites to increase storage space
  - Clusters are not disjoint, every satellite is member of 7
- Satellites with same role/color exchange cache information
  - Called **same-color-merge**



Cache performance in a satellite constellation

- Purple lines: Distributed cache based on **hash values (modulus 7)**
- Green lines: Included **same-color-merge**

Green line shows faster training important due to diversity of query term distribution



Conclusion

- Cache performance is essential in a SIN
  - Satellites will always be a “slave” replica
  - Well suited for cooperative caching, with some adaptations
  - Performance in a SIN is **nearly the same** as with a “stable” term distribution
- What else need to be solved for a SIN to be useful?
  - Handover / state-transfer
  - Robustness in the presence of faulty satellites
  - Protection, authentication and sharing
  - Routing around densely populated regions
  - Application design patterns (delay-sensitive vs. delay-tolerant activities etc.)