Transfer of Session State Between Satellites in a Space Information Network

dr. Anders Fongen, March 2023
Norwegian Defence University College, Cyber Defence Academy, Lillehammer
email: anders@fongen.no

INTERNET2023, Barcelona, Spain

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
  - 6 years in military engineering education (Associate Professor)
  - 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
    - Predictability of positions, links, routes and workload
    - Long idle periods (due to inhabited surface) mixed with traffic peaks
- Viewed as a problem of *Distributed Computing*
    - *having a set of distinct properties*

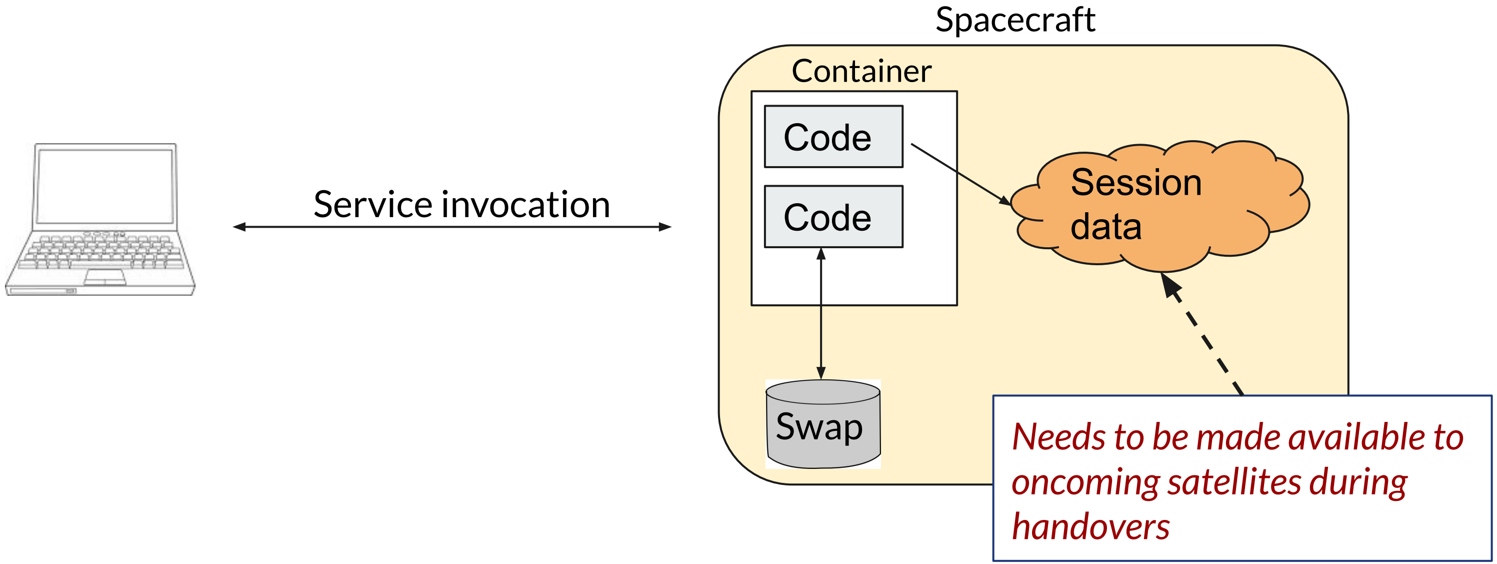
What is a SIN (Space Information Network)?

- 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*

Service deployment in a SIN

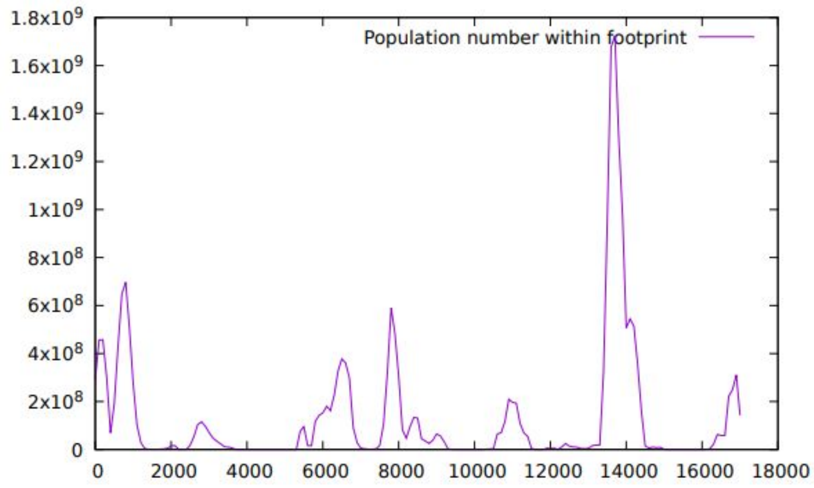
- A service need code segment, local data and *session data*
  - code segment is common to all service clients, and “immutable”
  - session data is separate for each client, and is often updated
- During handover from one satellite to the next
  - code and session data must be made available on the oncoming spacecraft.
    - code segment may be deployed on new satellites, proactively or on-demand
    - session data must be *made available* to new satellite (not necessarily copied)
      - different methods for this task will be subject to further discussion.

Code and session data in a service provider

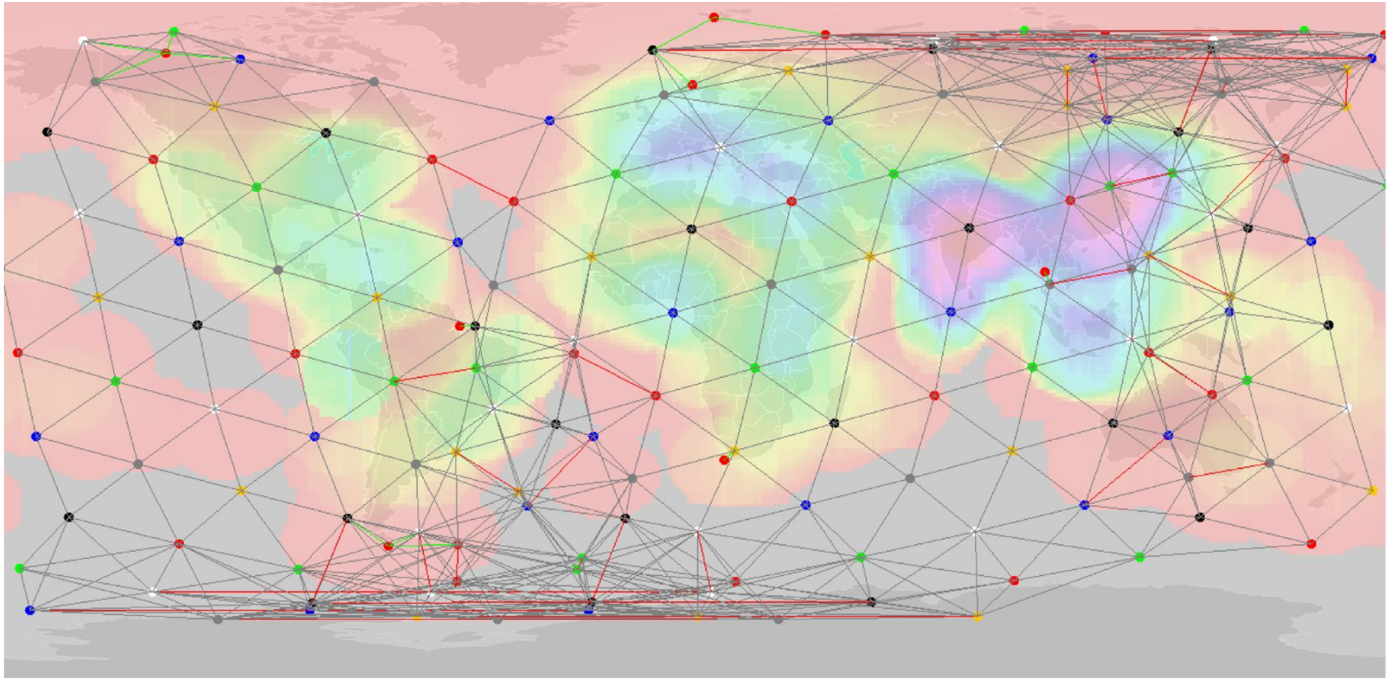


Workload prediction - demographic data

- During orbit, a satellite is idle most of the time
  - During which it can offload the busy satellites, like ***storing session data elements***
- Experimental results will follow….



Population “heat map” from satellite footprint



Methods for session data management

1. Keep one copy of session data in a stable and reachable location (e.g., on the ground)
    - defeats the purpose of a SIN
2. Copy entire session data to the oncoming satellite
    - reduces access latency, but creates much copying traffic
    - creates uneven workload of satellites and links (due to population distribution)
3. Copy session data elements to oncoming satellite **on demand**
    - creates a balance between access latency and copying traffic

Investigating alternative 3: on demand copy

Access operations to session data elements are assumed to follow a *scale-free distribution*, where a few elements are often accessed, others less often. Inversely proportional to their *rank*.

$$f = \frac{a}{r}$$

Where a is given a value so that

$$\sum \frac{a}{r} = 1$$

On-demand copying of session data elements will reduce the number of copy operations.

We will arrange the session data as pages of *virtual memory*, consisting of a *page table* and a number of *pages frames*

Session data as *distributed virtual memory*

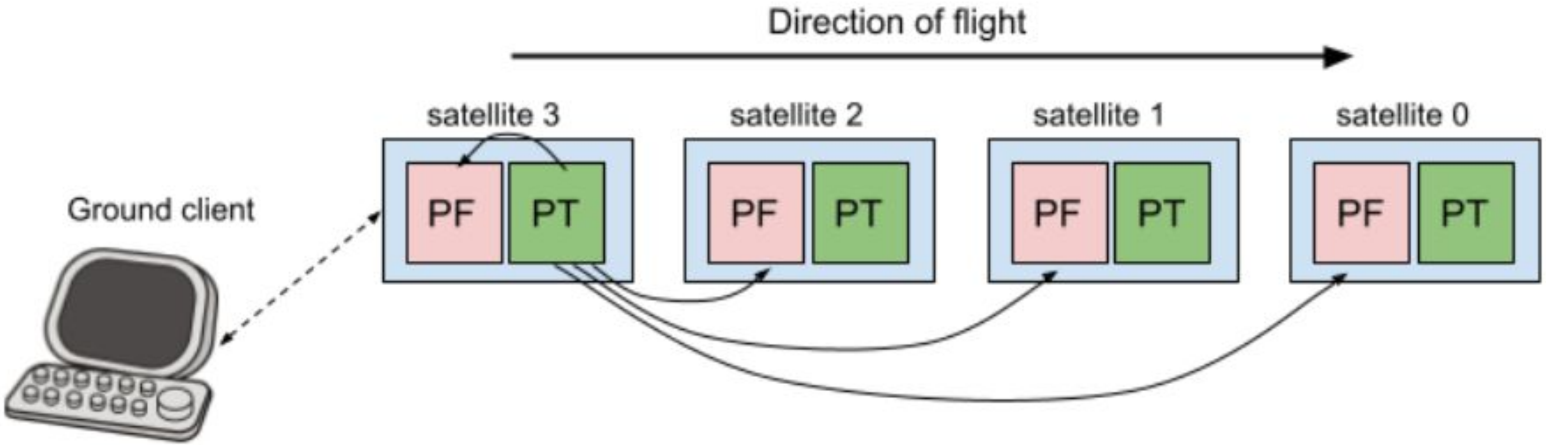


Figure 3. Pages are distributed in PFs along the trail of satellites, and referenced from a single PT.

Performance of on-demand copy

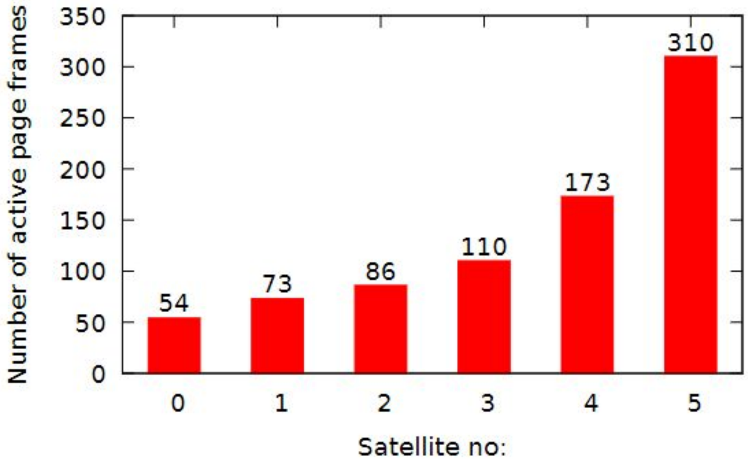


Figure 4. The distribution of session state pages after 5 handover operations.

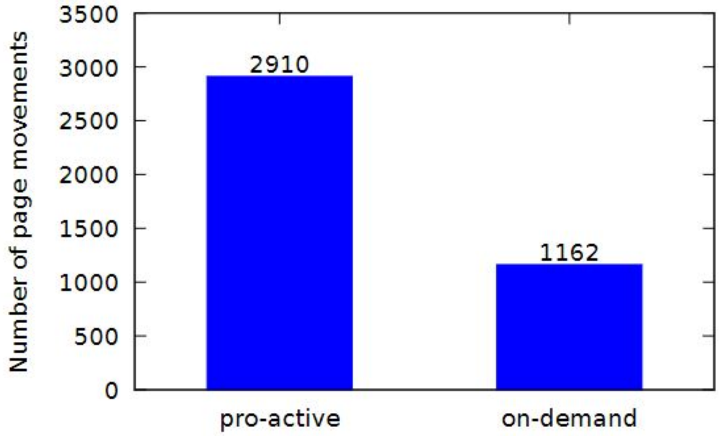


Figure 5. The number of session state page movements for pro-active and on-demand method

Conclusion

The problem: **How to maintain session data for stateful containers in a SIN**

- Elements of session data are accessed according to a **scale-free distribution**
- Session data in a satellite is organized as a **virtual memory** segment
  - Elements are fetched on demand subject to **page faults**.
  - Elements never accessed are left in the trail of satellites
- On-demand copying of elements generate **60% less** network traffic.

Thank you for your attention, any questions?