Data Sharing Services in a Space Information Network

Anders Fongen
Norwegian Defence University College (FHS)
Lillehammer, Norway
email: anders@fongen.no

***Abstract*—For collaboration services in a Space Information Network, services and data structures for sharing are essential. The organization and placement of resources for that purpose must consider the semantics of the sharing operations, like discovery, transactional properties, update ordering, and update notification. These aspects are analyzed and discussed in the manuscript, and the more realistic sharing mechanisms are also evaluated through simulation experiments.**

Keywords*—*LEO satellites; space information networks; data sharing; mobile computing.

I. INTRODUCTION

The term *Space Information Network* (SIN) describes a set of satellites that cooperatively offer services for information processing and sharing, as well as traditional communication services. SIN is regarded as a natural evolution of satellite services, from radio mirrors in geostationary orbit to Low Earth Orbit (LEO) constellation for communication services (e.g., Iridium) [1][2].

In a series of previous publications, different aspects of SIN operation (architecture [3], security [4], cache management [5], routing [6], and state management [7]) have been addressed. This article will focus on the design of data sharing mechanisms, their semantics, resource placement and optimization.

For the sake of successful collaboration services in a SIN, data sharing mechanisms are strictly necessary, and should not be left to ground-based services. An important advantage of a SIN is the potential for very low latency, which is best maintained by a sharing service offered by the satellite network itself.

Application clients on the surface, denoted C_a, will connect to any satellite overhead, which serves as an *application server*, S_a. When in need for access to shared data, S_a will communicate with the sharing instance S_s. Three tiers are thus involved, and C_a will never make direct contact with S_s. Please observe that while the C_a to S_a connection is a *link*, the S_a to S_s connection can have multiple hops. These relations are illustrated in Figure 1.

Data sharing in a SIN will benefit from the predictable properties of a satellite network. The position of every satellite can be computed by anyone at any time, as well as inter-satellite link availability and the population density inside a satellite's footprint. These properties alleviate the need for a discovery service and allow the resource management to anticipate the periods of high and low traffic intensity from surface clients.

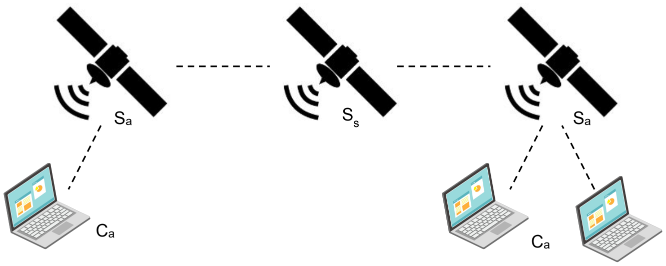

Figure 1. The relation between the application clients (C_a), application servers (S_a) and shared data servers (S_s).

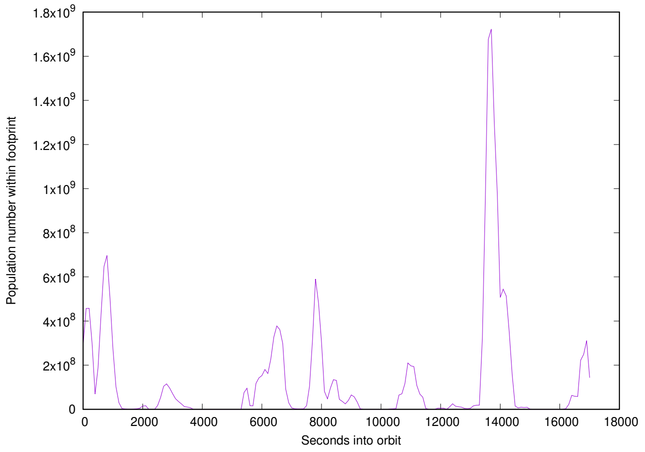

Figure 2. The population number inside the footprint of a satellite during three subsequent orbits.

The population distribution of the planet is highly uneven. A satellite (serving the S_a role) in orbit will expect great variations in the workload offered by C_a: short-termed peaks of high traffic intensity between longer periods of little activity. Figure 2 shows the population density variations during three orbits at 500 km altitude. This property represents a scalability problem on one hand, and an opportunity for improved resource planning on the other.

Another property of any LEO network is the need for frequent handover of C_a to S_a link, including migration of those resources on which C_a depends. Session state and service endpoints need to shift to other S_a in order to maintain the communication to the C_a [7].

The perspective of the presented analysis is that from Distributed Computing. Technical and physical properties of satellites related to energy management, antenna design, modulation, coding, jamming resistance etc., are not taken into consideration.

In the analysis following in this paper, the assumption is made that the access operation from S_a to the data elements in S_s is *not* uniformly distributed. The access frequency follows a *Scale Free Distribution* model (SFD), in which the access frequency of an element is inversely proportional to it *rank*.

The remainder of the paper is organized as follows: Section II will discuss typical and essential properties of a sharing service, and Section III will briefly present the software simulation model used in the experiments. Alternative methods for migration of shared data during handover operations are discussed in Section IV, while an optimization experiment for placement of a shared service is presented in Section V. Experimental results regarding path cost for connections are presented in Section VI. Finally, the paper lists its conclusions in Section VII.

II. DATA SHARING SEMANTICS

Operations on data elements in S_s involve critical regions which must be protected to avoid race conditions and update conflicts. Furthermore, there are semantic properties related to access and update operations which should be considered for implementation:

a) Transactional atomicity and isolation: Assumed implemented in the traditional ACID way. If used correctly, it protects the data element against update conflicts. It requires a network-wide mutex mechanism, which is a dangerous thing to use since S_a can crash and restart any time.

b) Ordering semantics: Guarantees the observed ordering of updates received from other shared data collections. This is a well studied problem, discussed, e.g., in [8]. The so-called *causal ordering semantics* strikes a good balance between implementation complexity and usefulness to a range of communication patterns (through the use of, e.g., Lamport clocks [9]). Other ordering models, like the FIFO ordering and the less formal “eventual consistency” are easier to implement but also less useful.

c) Update notification: An event notification system whereby S_a can listen for notifications from update operations on the S_s allows for interesting application patterns. The *observer/observable* and the *model-view-controller* patterns are well known to any programmer, they allow multiple S_a to obtain an (eventually) consistent view of the shared data without the need for continuous polling of its state.

d) Relational data base: A well known and mature organization of a data set expressed through relational algebra, with excellent support for transactional properties, security and redundancy. May well serve as a service endpoint in S_s, but does not lend itself well to frequent migration.

The sharing models listed above have different semantic properties, and varying ability to operate under the circumstances found in a SIN. An analysis of how well they may adapt to the *mobility properties*, *S_a group dynamics*, etc., is presented in Sections II-A and II-B.

A. Mobility properties

For a C_a, an S_a will stay within line of sight for approximately 15 minutes (assuming a LEO satellite constellation) after which a handover operation needs to take place. A handover implies that all the resources necessary to uphold a continuous service must be migrated to the new satellite. In addition, resources referenced in other satellites will also move in the same fashion, and a handover of these resources may be necessary in order to maintain a short communication path.

The mobility pattern of satellites is completely predictable, which means that any two identified satellites can predict if they are within radio range of each other. Link discovery is not necessary, but the connecting part will need to learn the ID of the other satellite for this calculation to take place [6]. Once the initial connection has been established, it can be used to notify the peer of upcoming handover operations.

The handover operation for a S_s may find it unnecessary to migrate all shared data elements to the next satellite, but will need to establish the service endpoint in the new satellite and notify all client S_a about the new endpoints. Shared data elements may be left behind and fetched on-demand, which will be described in Section IV.

B. Client group dynamics

Associated with an S_s is a group of S_a instances. The membership of this group is constantly changing as the S_a’s need for shared data access emerges and ends.

An S_a will need to know where the needed S_s is, also during and after a handover operation. If the S_s need to send update notifications to the S_a, some form of group membership protocol must be in operation which will consume a portion of the communication capacity. A better arrangement for group membership handling will be presented in Section VI.

C. Shared memory or service interface

Two possible methods for S_a access to S_s are (1) *shared memory* or (2) *service interface*. Method (1) allows S_a to access shared data like memory cells in the computer, for either write or read. In order to protect read-test-write sequences from race conditions, distributed mutexes must be in place. A distributed mutex carries the risk for deadlocks in the case where S_a crash with acquired mutexes. Also, the method is unable to offer any consistency, event notification or ordering guarantees.

The list of properties in this section strongly suggests a service oriented interface for the shared data in S_s. A service implementation allows both simple and composite operations to take place in a threadsafe, synchronized, protected and reliable manner. Event notifications can also be offered from a service interface, although long-termed (asynchronous) service invocations need to consider handovers and change of IP addresses during the invocation.

For the rest of this manuscript, S_s access is assumed to be implemented through a service interface, not as direct access to memory cells.

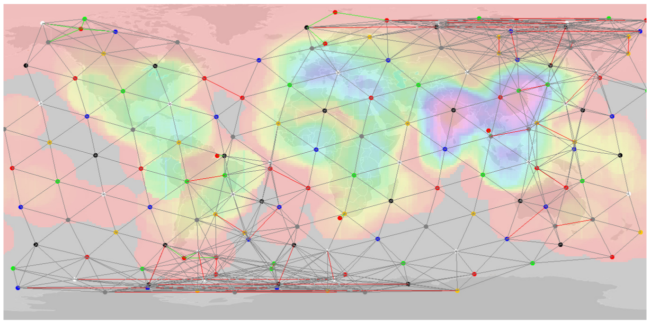

Figure 3. Screenshot from the satellite constellation model.

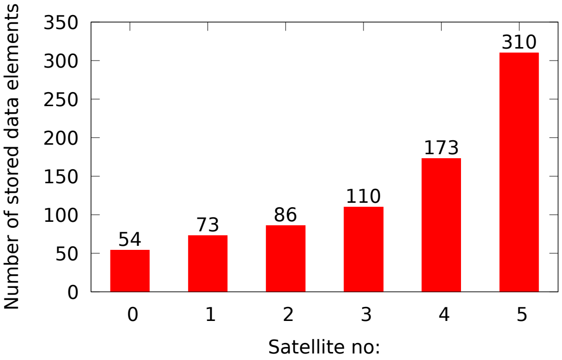

Figure 4. The distribution of shared data elements after 5 handover operations.

D. Shared data access pattern

The shared data elements in S_s are not likely to be accessed in a uniform manner, i.e., with the same access frequency, but more likely to be accessed according to a *Scale Free Distribution* model (SFD). The SFD predicts usage and access patterns for a range of human activities and natural phenomena [10]. In this particular use case, the data elements are *ranked* according to their access frequency. Assuming SFD, the relative frequency (f) of accesses to an element is expected to be inversely proportional to the rank (r) of that element. Applied to this use case, SFD predicts that the most frequently used element will be accessed twice as often as the second most frequently used element, three times more often than the third most frequently access element, and so on. Mathematically, this may be expressed as

$$f = \frac{a}{r} \tag{1}$$

where a is given a value so that

$$\sum_r \frac{a}{r} = 1 \tag{2}$$

III. THE SOFTWARE MODEL

The results presented in this article are based on a software simulation of a satellite constellation. A screenshot from the model is shown in Figure 3. The constellation consists of 150 satellites at 500 km altitude. The colored backdrop in the figure indicates the population density inside the satellite footprint at a given location, based on gridded population data from NASA [11]. This data set has also been used to calculate the graph in Figure 2.

IV. MIGRATION PATTERN FOR SHARED DATA

The actual location for S_s needs to be established, and the most obvious method would be to store the shared data elements inside the satellite which offers the S_s service interface. In this case, the entire data set will be migrated during a handover of the service interface.

The assumption was made in Section II-D that the pattern of access operations follows a *Scale Free Distribution* [10] which allows for a more scalable design: After a handover operation of the S_s service interface, the shared data elements are not migrated until they are referenced by a call from a S_a. The elements which are never referenced during the following service period are then left behind by *two* hops at the next handover operation, etc. The elements that are never again referenced will therefore not consume any link capacity. Although applied for a different use case, the results obtained in [7] use similar mathematics and will be presented here as a simulated result for on-demand migration of shared data elements.

The distribution of shared data elements has been simulated with these parameters:

- 5 handover operations, involving 6 S_s satellites
- 1000 S_s accesses from S_a to each satellite
- There are 1000 shared data elements in total
- Shared data elements are accessed according to SFD

The resulting distribution is shown in Figure 4. The sum of all numbers (806) shows that far from every shared data element were ever accessed, and existing elements remaining in storage of previous satellites indicate that they were never accessed since that satellite's time of service.

The number of link traversals used for migration of data elements during the course of the simulation is chosen as an indicator of the scalability properties of the arrangement. Therefore, the number of single hop movements of elements will be analysed under the scenario described in this section. The resulting numbers from the proactive and the on-demand migration method are compared and reported.

The distribution of the shared data elements across the current and past S_s satellites was measured just before a handover operation, after 1000 access operations. The numbers are shown in Table I. From these numbers, it is possible to calculate the total number of element movements across inter-satellite links during the scenario of 5 handover operations with 1000 access operation between each.

For the proactive element migration method, the total number of link traversals is 2910 (the sum of the 5 first numbers in the "total" column). For the on-demand method, the total number is 1162. This means that the on-demand method consumes only 40 % of the communication capacity required by the proactive method.

TABLE I
DISTRIBUTION OF SHARED ELEMENTS ACROSS S_s SATELLITES BEFORE EACH HANDOVER.

satellite total	0	1	2	3	4	5
336	336					
499	172	327				
608	118	168	322			
704	91	113	157	343		
763	67	94	108	164	330	
806	54	73	86	110	173	310

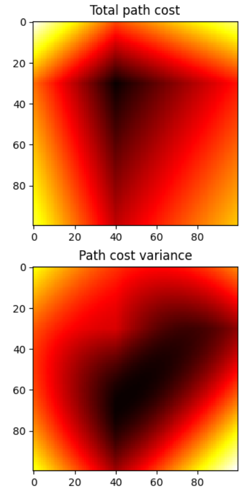

Figure 5. The distribution of total path cost and its variance for 5 S_a inside a grid of 100x100 possible locations for the S_s.

V. BEST PLACE FOR THE SHARING SERVICE

Given that there is a number of S_a satellites in need of cooperation over a S_s service, there is an optimal position for S_s based on one or more criteria.

For client satellite p, the access frequency to the shared service is f_p and the path length to the service is d_p. The chosen criteria are:

1) The total number of link traversals, i.e., $\sum d_i \cdot f_i$, should be minimized.
2) The network load offered by the clients should be evenly distributed, i.e., $\mathrm{Var}(d_i \cdot f_i)$ should be minimized.

A quick look into these criteria is shown in Figure 5 as a heat map. The black color represents the lowest number, white color the highest. The S_a nodes are placed in each corner with access frequencies (f_p) 1,2,4 and 8, and a fifth client in position (30,40) and $f_p = 15$. Each pixel in the heat map represents a candidate placement for S_s and the color the corresponding sum and variance for the path cost for S_a to S_s communication.

As shown in the heat map, the lowest total path cost is obtained with the S_s located near the S_a with highest f_p, in this case the client in position (30,40). The heat map also shows that this position does not give the lowest variation, which is found a little to the "south" of it.

VI. CONNECTIVITY AND PATH LENGTH

Beside migration of shared data elements and optimal placement of the S_s, the conditions for the communication between the S_a and the S_s should be investigated.

Possible strategies for routing inside the satellite grid in a SIN have been studied in [6], where the hypothesis was that the forwarding path should aim to involve less busy satellites. Also important, the study showed that the grid is never partitioned, so one S_a will always be able to communicate with a S_s, regardless of their positions. This property is also shown in Figure 3. The path cost for general routing shows a significant improvement with that approach, but also large fluctuations. There is consequently no need for a separate routing mechanism for shared data access, but a *discovery service* for S_a to find the S_s at the beginning of a session is needed. During a S_s handover, the server can leave a "breadcrumb" which points to the new location, for the subsequent redirection of the S_a.

VII. CONCLUSION

Any collaboration between a group of C_a will require coordinated access to a body of shared data. This paper has investigated four aspects of a sharing mechanism: Service semantics, migration of shared data elements, placement of the shared service endpoint, and routing methods for communication between S_a and S_s.

The optimization methods suggested during the course of the paper are based on the special properties of a SIN operation: Predictability and uneven population density.

This is a part of an ongoing feasibility study and there are still many details in need for a detailed study, which will be the focus for further research effort in the field of SIN operation.

REFERENCES

[1] S. Briatore, N. Garzaniti, and A. Golkar, "Towards the internet for space: Bringing cloud computing to space systems," in *36th International Communications Satellite Systems Conference (ICSSC 2018)*, 2018, pp. 1–5.
[2] L. Bai, T. de Cola, Q. Yu, and W. Zhang, "Space information networks," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 8–9, 2019.
[3] A. Fongen, "Application services in space information networks," in *CYBER 2021*. Barcelona, Spain: IARIA, Oct 2021, pp. 113–117.
[4] A. Fongen, "Trust management in space information networks," in *SECURWARE 2021*. Athens, Greece: IARIA, Nov 2021, pp. 14–18.
[5] A. Fongen, "Cooperative caching in space information networks," in *INTERNET 2022*. Venice, Italy: IARIA, May 2022, pp. 1–5.
[6] A. Fongen, "Population-based routing in leo satellite networks," in *MOBILITY 2022*. Porto, Portugal: IARIA, June 2022, pp. 1–4.
[7] A. Fongen, "Transfer of session state between satellites in a space information network," in *INTERNET 2023*. Barcelona, Spain: IARIA, March 2023, pp. 1–4.
[8] A. S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2007.
[9] L. Lamport, "Time, clocks, and the ordering of events in a distributed system." *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, jul. 1978.
[10] A.-L. Barabasi, *Linked: How Everything Is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. Plume Books, April 2003.
[11] "Gridded population of the world v.4.11," [Online; retrieved 15-Aug-2023]. [Online]. Available: https://sedac.ciesin.columbia.edu/data/collection/gpw-v4/sets/browse