Space-Based Wireless Sensor Networks: Design Issues

Tanya Vladimirova, Christopher P. Bridges, Jean R. Paul, Saad A. Malik and Martin N. Sweeting
Surrey Space Centre
Department of Electronic Engineering
University of Surrey
Guildford, Surrey.
+44 1483 300800
{t.vladimirova, c.p.bridges, j.paul, s.malik, m.sweeting}@surrey.ac.uk

Abstract—This paper is concerned with a satellite sensor network, which applies the concept of terrestrial wireless sensor networks to space. [1,2] Constellation design and enabling technologies for picosatellite constellations such as distributed computing and intersatellite communication are discussed. The research, carried out at the Surrey Space Centre, is aimed at space weather missions in low Earth orbit (LEO). Distributed satellite system scenarios based on the flower constellation set are introduced. Communication issues of a space based wireless sensor network (SB-WSN) in reference to the Open Systems Interconnection (OSI) networking scheme are discussed. A system-on-a-chip computing platform and agent middleware for SB-WSNs are presented. The system-on-a-chip architecture centred around the LEON3 soft processor core is aimed at efficient hardware support of collaborative processing in SB-WSNs, providing a number of intellectual property cores such as a hardware accelerated Wi-Fi MAC and transceiver core and a Java co-processor. A new configurable intersatellite communications module for picosatellites is outlined.

TABLE OF CONTENTS

1. INTRODUCTION

This paper is concerned with space-based wireless sensor networks (SB-WSNs) consisting of very small satellite nodes flying in close formations. The main idea of SB-WSNs is that rather than having a single large expensive satellite to achieve the goals of a mission, a large number of inexpensive (mass producible) satellite nodes are deployed in a formation to achieve the same goals.

There are some important astro-dynamics and engineering research challenges to enable formations in low Earth orbit (LEO). Perturbations have been shown to reduce the lifetime of local satellite clusters and constellations, so an implementation of the recent Flower constellation model [1] has been investigated and adapted for a LEO mission scenario. Secular drift can be mitigated by using a more equatorial inclination and atmospheric drag can be mitigated via a higher eccentricity. Geometric shapes can be formed to produce ‘flower’ shapes with the ‘petals’ giving angular requirements of each satellite position. Current simulations envision that a LEO distributed mission is feasible using the Flower constellation model. Scenarios have been explored where picosatellite constellations drift in and out of inter-satellite link (ISL) length between a range of 400 km and 100 km, presenting a dynamic and often ‘disconnected’ environment. The need for an ad-hoc and autonomous distributed computing platform to enable collaboration via ISLs is obvious in this environment for enabling future distributed satellite missions.

Future spacecraft are envisioned as autonomous, miniature, intelligent and massively distributed space systems. The concept of satellite sensor networks can be applied to many space missions [2, 3]. Some examples include:

- realising co-orbiting assistants/ inspectors of larger mother ships;
- providing continuous Earth coverage for multipoint remote sensing, monitoring or communications at low cost in LEO;
- providing continuous communications for multiple low-powered surface vehicles around the Moon, Mars and other planets or asteroids.

Space weather is associated with many of the anomalies detected on spacecraft [4, 5, 6]. In LEO spacecraft is particularly vulnerable when it passes the poles - home to the auroral ionized belts and the South Atlantic Anomaly (SAA), where ionized particles come very low into the atmosphere. Service outages of the satellite navigation system due to solar storms are a cause of great concern [7]. Distributed networked small satellite missions could be used to study the impact of solar storms on Earth’s magnetosphere and ionosphere increasing the spatial and

[1] 978-1-4244-3888-4/10/$25.00 ©2010 IEEE
[2] IEEEAC paper#1498, Version 1, Updated 2009:11:01

temporal resolution and providing continuous in-situ measurements. Replacing a group of sensing satellites, which operate separately in their own local vicinity, by networked satellites operating in a distributed fashion will also increase the science return per dollar ($) as envisioned in DARPA's F6 project [8].

This paper is organized as follows. Section 2 introduces the distributed satellite system constellation scenario based on the flower constellation. Section 3 focuses on the design issues of a space based wireless sensor network in reference to the OSI layer stack. Section 4 details a system-on-a-chip computing platform and agent middleware for distributed processing in SB-WSNs. A new configurable ISL communications module for picosatellites is outlined in Section 5. Section 6 concludes the paper.

2. MISSION CONSTELLATION SCENARIO

A distributed satellite system requiring intersatellite links could be formed for a number of missions. For each mission, specific orbits would be required to meet the mission goals, taking advantages of intersatellite links. These missions are summarized in Table 1.

Table 1. Constellation Orbital Characteristics and Applications

Const.	Characteristics	Applications
String-of-Pearl	Polar/ sun-synchronous orbits Predictable connection periods Limited mobility	1. Earth/ space observation 2. Communication 3. Global positioning/ navigation 4. Science
Flower	Elliptical orbits Predictable connection periods Known mobility patterns	1. Multi-point atmospheric/ space weather monitoring 2. Distress beacon monitoring 3. Experimental orbits for Earth observation, communication and positioning
Cluster	Similar orbits Unpredictable connection periods Medium/ high mobility. Unknown patterns	1. Hardware Fractionation 2. Multi-point atmospheric/ space weather monitoring 3. Earth observation, communication and positioning

Table 1 highlights some of the orbit characteristics for each of three constellation designs – string-of-pearl, Flower constellation and satellite cluster. Depending on the mission needs and orbital characteristics, parameters of the intersatellite communication, whether for brief or long periods, can be predicted.

2.1. The Flower Constellation

The *Flower constellation* set provides stable orbital configurations, which are suitable for micro- and nano-satellite missions. Applications proposed and initially investigated include GPS missions, reconnaissance, two-way orbits, multiple science missions and planetary exploration [9]. Upon closer investigation, there are some distinct features including [1]:

- The constellation's axis of symmetry coincides with the spin axis of the Earth.
- Each satellite has the same orbit shape (anomalistic period, argument of perigee, height of perigee and inclination).
- Satellites are equally displaced along the equatorial plane to complete the constellation using the right ascension of the ascending node (RAAN), true anomaly or mean anomaly.

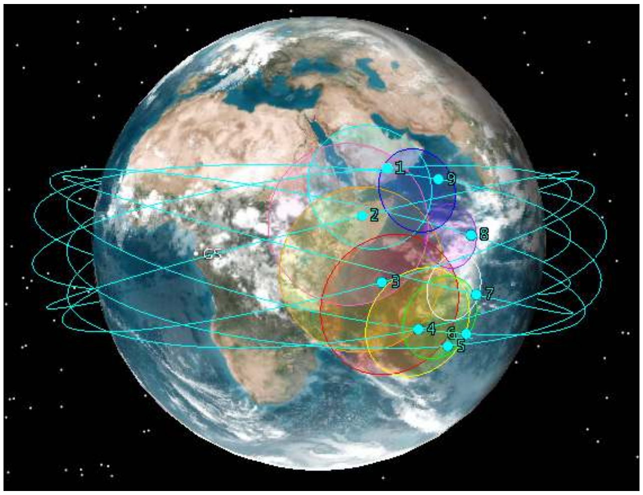

Figure 1. Flower Constellation

Previous research applied the Flower constellation to low Earth orbit (LEO) for a set of 9 picosatellites giving constant and predictable ranges from 100 km to 400 km between neighbouring satellites [10]. Unlike polar orbit constellation scenarios, the Flower constellation with a more equatorial inclination ensures that the satellites will drift together along the Earth's equator; keeping them in formation for a much longer without the need for orbit maintenance. The proposed Flower constellation in the equatorial plane is particularly promising for the launch of picosatellites (mass < 1 kg) or nanosatellites (mass < 10kg).

Simulations were carried out using AGI's High Precision Orbital Propagator (HPOP) in Satellite Toolkit (STK) [11]. Figure 1 provides an image of the Flower constellation of 9 picosatellites and Table 2 describes the design parameters used.

Table 2. Satellite and Orbital Properties for the Flower Constellation

Satellite Properties	Value
Mass, m	1 kg (picosatellite)
Volume	10 cm^3
Cross sectional area, a	20 cm^2 (tumbling)
Co-efficient of drag, C_D	2.2 (flat plate model)
Atmospheric density, ρ	2.961 x 10^{-13} kg/m^3
Ballistic co-efficient, B	$B = \frac{1}{2} C_D \frac{a}{m} \rho = 5.92 \text{ x } 10^{-16}$
Orbit Properties	**Value**
Apogee altitude, h_a	1598 km
Perigee altitude, h_p	686 km
Inclination, i	165 °
Right ascension of the ascending node, Ω	Satellites 1-9: 0, 40, 80, 120, 160, 200, 240, 280, 320 °
True anomaly, θ	Satellites 1-9: 0, 53.54, 98.12, 134.1, 165.2, 194.8, 225.9, 261.88, 306.46 °

2.2. Flower Constellation Design Issues

When looking at any mission aiming to use intersatellite links, important orbital factors to consider are relative range/ speeds between satellites, the ISL access opportunity, and the ground-link access opportunity. The *access time* between each satellite is proposed as the best metric to predict distributed collaboration. The access time is the time for two picosatellites to communicate between each other dependent on a set range. The communication range of 400 km is chosen in this modelling study, which is assumed to give sufficient collaborative opportunity. Figure 2 shows the access time for the constellation in Figure 1 showing picosatellites drifting in and out of range at different times.

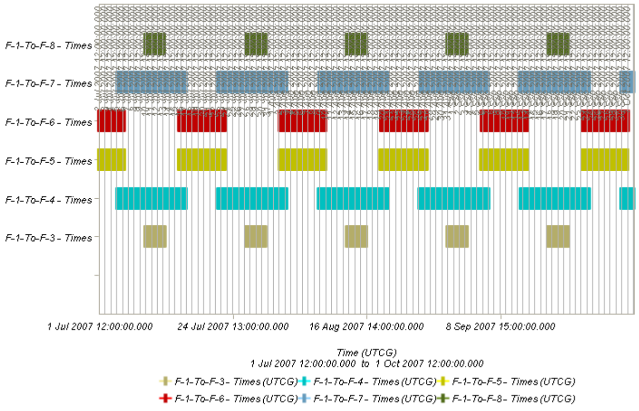

Figure 2. Flower Constellation Access Times for Nine Picosatellites

Access times between picosatellites range between 3 days to 14 days dependent on the main *sink satellite*. The sink picosatellite is the master satellite that communicates to ground and can be used for controlling distributed operations.

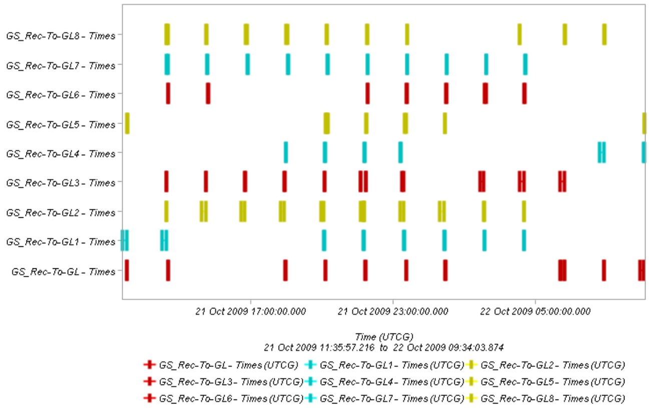

Figure 3. Groundstation Access Times for the Flower Constellation

The sink satellite needs to be chosen because if all satellites tried to communicate to ground, the link would be over-subscribed (assuming one operational frequency). For example, Figure 3 shows that between 3 to the maximum 9 satellites could be in view at any one given time. The simulations presented in Figure 2 and 3 suggest predictable and repeating patterns for both intersatellite and groundlink connection periods. However, it has to be noted that the simulation results are as close to the true orbits as good the force models for predicting the orbits are.

In order to achieve the initial conditions of the Flower constellation the satellites must be positioned in a certain way during or after deployment. Intersatellite communication capability could help to overcome difficulties in identifying positions of individual satellites and predicting their orbits after deployment.

3. Network Design Issues

As discussed in Section 2.2, spacecraft crosslink communications are affected by orbital dynamics, which impose a number of difficulties and restrictions such as variable inter-satellite ranges and speeds, variable ISL access for distributed operations, etc.. To investigate these problems we use the Open Systems Interconnection (OSI) networking scheme [12]. The functionality of the OSI layers can be implemented in hardware or software, as shown in Figure 4. This section will discuss the different design issues that are found at each layer.

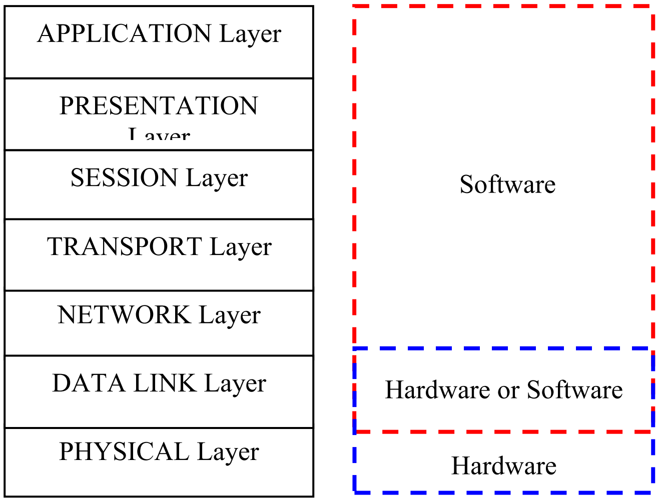

Figure 4. OSI Layers and Implementation Methods

3.1. Physical Layer

Radiation is one of the primary environmental hazards in space affecting on-board electronic components and propagation of communication signals [13, 14].

Ground communications in picosatellite designs are in the VHF and UHF bands. VHF frequencies in the range of 30 to 300 MHz normally pass through the ionosphere with effects such as scintillation, fading and Faraday rotation etc. However in times of intense solar cycles, VHF signals can be reflected back causing multi-path effects. Cases observed during peaks of cycle in 1957-58, Cycle 21 in 1980, and Cycle 22 in 1990 [14]. VHF signals can also get reflected by auroral strips in the extreme solar activity. Between 300 MHz and 3GHz, in which S and L band lie, severe disruptions are possible during a solar storm [7] which could affect intersatellite link communications.

Global positioning signals (GPS) are deemed to be an essential tool for orbit determination and navigation on board constellation satellites. Solar storms are known to cause synchronization and phase lock errors in GPS receivers [7].

3.1.1. Radiation Effects: Communications Channel

The Appleton-Lassen formula is a well known propagation model for ionospheric propagation, which describes the complex refractive index of the medium. If the magnetic field is ignored then the real part of the refractive index ∂ is given as [15]:

$$\partial^2 = 1 - \left(\frac{f_N}{f}\right)^2 = 1 - k\left(\frac{N}{f^2}\right) \qquad (1)$$

where $k = 80.5$ is a constant, N is the electron density per cubic metre and f is the operating frequency in Hertz. The critical frequency of plasma is denoted as f_N.

It is important to note that N varies and its average value is 10^5 for altitudes up to 1000 km during daytime. The change in electron density affects the critical frequency and has been known to have caused reflections in frequencies above the critical frequency [15]. Since electron density is variable, a configurable and robust communications system is essential.

Other critical parameters affecting propagating waves at a given electron density are:

- Varying group velocity as well as phase propagation delay.
- Attenuation, caused by electron-neutron collisions.
- Refraction due to varying plasma density, causing multi-path effects.

3.1.2. Antenna Pointing and Power

Given the limited power resources on board picosatellites, adaptive techniques could be used to optimize power utilization. The relative velocity between satellites in different orbits varies with time. This results in a time-varying azimuth and elevation, and in addition places constraints on the antenna steering. Analytical modelling of ISLs for circular orbits is presented in [16, 17]. It is shown that the variation of the elevation is small, whereas the azimuth varies significantly [16]. The following is an expression for evaluation of the azimuth , ψ, [17]:

$$\tan\psi(t) = \frac{\left[\sin\theta \sin\left(\omega t + \alpha_2 - \frac{\pi}{2}\right)\right]}{\left\{\sin^2\left(\frac{\theta}{2}\right)\sin(2\omega t + \alpha - \pi) - \cos^2\left(\frac{\theta}{2}\right)\sin(\alpha_2 - \alpha_1)\right\}} \qquad (2)$$

where θ is the angle of separation between orbits, ω is the angular elongation, α_1 and α_2 represent the latitude of two neighbouring satellites 1 and 2, and α is the sum of α_1 and α_2.

An expression of the ISL length as a function of the azimuth is derived in which the expression for the azimuth in (2) above is substituted. A mathematical model for the power of the receiving antenna as a function of latitude is then developed substituting the ISL length expression in the Friis free space equation. Figure 5 shows calculated power variation of the receiving antenna for intersatellite communication in LEO circular polar orbits using that model.

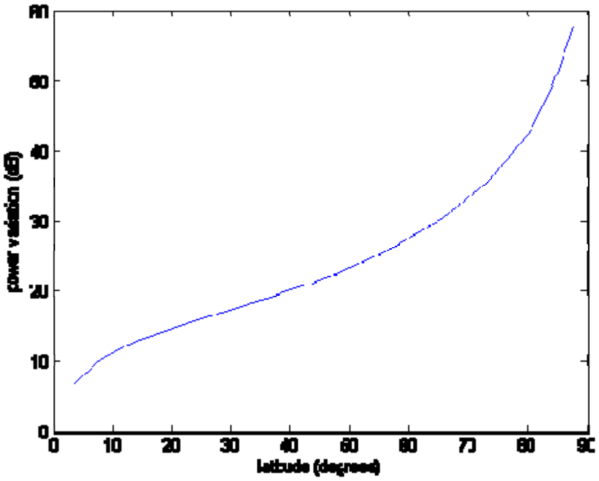

Figure 5. Power Variation with respect to the Latitude in the Southern hemisphere

It can be seen from the graph in Figure 5 that the power of the receiving antenna varies within 58 dB having a minimum at the equator and a maximum at the poles. This can be exploited for implementation of adaptive power control on board to reduce the power consumption varying the transmitter's antenna gain based on pre-calculated azimuth or latitude values.

3.2. Data Link Layer

Due to bandwidth scarcity in wireless networks, a common approach is to use a multiple access scheme to share the bandwidth of a communication link between several nodes. The link layer delimits groups of bits to form frames, and switches are used to dispatch frames to the correct node. A control mechanism called Medium Access Control (MAC) is used to manage the communication link. The MAC layer ensures that frames are delivered error-free, and adds addressing information to the transmitted frames.

Existing commercial lower layer protocols and their suitability for intersatellite communication in autonomous constellations are discussed in [18]. It is concluded that long propagation delays, appropriate data rates, and forward error correction mechanisms are features required for reliable space communications.

It has been shown that the terrestrial IEEE 802.11 wireless network standard can been adopted for intersatellite link design [2]. In the IEEE 802.11 protocol carrier sense multiple access is used by nodes to monitor when the communication channel is free. Before a station is allowed to initiate a transmission, it senses the channel to verify whether it is free for a predefined minimum period called Distributed Inter Frame Space (DIFS). If the channel is busy, a random backoff interval is calculated to determine the waiting time before the sending station tries to access the channel again. IEEE 802.11 is a terrestrial communication protocol with ranges in the order of a few hundred metres, however it could be scaled up for communications range of a few hundred kilometres in space [19]. It is proposed to extend the range by redefining the MAC layer's distributed inter-frame space [16]. Although suitable for environments where the nodes are fixed, in a mobile environment, such as LEO, the proposed solution is not sufficient. Two scenarios are calculated for DIFS settings corresponding to communications range of 15 km and 100 km, as shown in Table 3. It can be seen that if the nodes are 100 km apart (DIFS=355 µs), the throughput drops by a factor of 3 compared with the DIFS setting for a range of 15 km. This suggests that an adaptive determination of the DIFS value is better suited to the needs of SB-WSNs, requiring that the ISL range is known in advance, or some form of range prediction is implemented.

Table 3. Throughput vs. DIFS Settings

Range (km)	DIFS (µs)	Throughput (Mbps)
15	75	3
100	355	0.94

3.3. Network and Application Layers

In SB-WSNs the extreme mobility and intermittent connectivity will affect the network topology requiring that the network is capable of reconfiguration. Routing optimisation based on minimising the transmission power and associated delays is proposed in [20]. It is concluded that satellite network requirements include:

- Ad-hoc intersatellite networking capabilities for initial topology formation such as IEEE 802.11 (WiFi) or 802.15.4 (ZigBee).
- Adaptable and redundant ground-link communication schemes, i.e. main 'sink' to ground.
- Proactive and reactive topology schemes to account for any mobility or node loss.

3.3.1. Middleware

Distributed computing is typically enabled by *middleware*, a software layer offering services to connect software components across a network for integration or sharing computing resources. The same connectivity issues affect the quality of service (QoS) for different middleware functions. For example, when two nodes connect using CORBA [21] or Java [22], they often register their services for resource sharing functions. But if they shortly disconnect and reconnect, there are often naming errors in the service registration that could cause an exception crashing the software system. Additionally, in a client/ server communication scheme, the most typical distributed computing paradigm, when a server or sink satellite fails then the network operations are lost. The chosen middleware must be autonomous and tolerant to satellite node failures, intermittent connectivity, changing connection topologies, and registration errors; analogous to an extreme case of *mobile ad-hoc networks* (MANET).

The application layer is mission and payload dependent, involving store and forward data transmissions with varying data sizes, which may require different communication schemes [10]. Higher rate data, such as payload data are suited to the Client/ Server communication scheme, while lower rate data would benefit from using the Peer-to-Peer (P2P) communication scheme. This can be telemetry, location or velocity changes such as "byte" size payload data (GPS, science payload measurements) & network management data (e.g. pinging). Future needs and applications for distributed operations, autonomy and artificial intelligence should be considered too based on current terrestrial software systems. Ideally, the management and payload data sizes transmitted across any channel (either the ground link or ISL) should be minimized as much as possible to reduce the power overhead of communicating.

4. Distributed Computing Platform Design

The work presented in this section is related to the computing support for data processing and communication at the SB-WSN node level. The implementation approach is based on hardware acceleration in the form of intellectual property (IP) cores for a system-on-a-chip (SoC) design [23]. The SoC uses the SPARC V8 LEON3 processor [24] and the AMBA2 bus [25]. Details are given about the development of two hardware accelerators - a WiFi transceiver and Java processor, and dedicated agent middleware.

4.1. Wireless Transceiver Core

The WiFi transceiver [26] is intended to operate in a mobile environment in which an adaptive DIFS will be used for range extension. Some of the IEEE 802.11 MAC layer functionality requires strict timing constraints. For instance, when a node receives a control signal, such as CTS, the data packet should be sent within a period of 10 µs called short inter-frame spacing (SIFS). Therefore, the MAC layer timing-critical functionality is implemented in hardware. However for ease of reconfiguration, a key function being considered is the communication range prediction via software which will implement the programming of the DIFS. Thus a hybrid hardware/software approach is employed to comply with the timing constraints.

The MAC is implemented as a hardware accelerator and the LEON3 processor is used to run software applications, interfacing the upper layers of the communication stack with the IEEE 802.11 protocol. The hardware accelerator implements a WiFi transceiver written in VHDL which contains functions such as 'byte by byte' processing in both receive and transmit directions, CRC generation for error detection purposes, signals to indicate successful transmissions, and reception.

Due to the asynchronous nature of communications in IEEE802.11 based networks, a mechanism for direct write from the receiver to the memory is required. As a result a direct memory access (DMA) core capable of controlling data transfer between the memory and the wireless transceiver is added to the design, shown in Figure 6. The DMA core has 32 channels to support up to 32 peripherals, and each channel has a number of registers allocated in the memory-mapped IO. An arbiter is placed within the DMA to give access to the component with highest priority. The registers are configured via the APB bus and are used to provide a set of functionalities to each component connected to the DMA. The registers allow to store information such as start addresses of the memory and the peripheral that require exchanging data, the data transfer size, byte counter. In transmission mode, the processor sends a signal to initiate data transfer using a register in the DMA; this involves moving data from the memory to the IEEE physical layer. In receiving mode a request signal is sent from the transceiver to the DMA to transfer data to the memory by bypassing the processor. Also when there is an error in the transmission a register is used to signal to the processor the type of error.

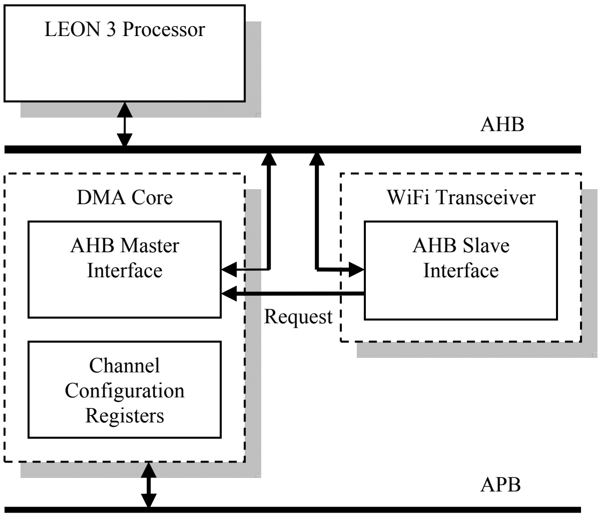

Figure 6. Wireless Transceiver Core Architecture

The MAC layer is divided in two parts. The transmitter state machine selects the correct sequence of packet type (control or data) and is responsible for CRC generation and forwarding data byte by byte to the physical layer. The receiver state machine monitors the carrier, collects data byte by byte, performs CRC and transfers data to the memory. The MAC interacts with the physical layer through an interface as shown in Figure 7.

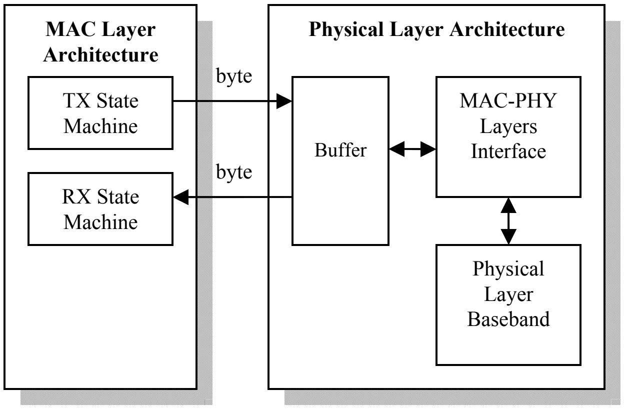

Figure 7. MAC layer 's Interface with Physical Layer

The MAC-Physical interface appends information such as preambles for packet detection, the data rate, modulation type and duration of data transfer. In order for the transceiver to meet IEEE 802.11 specifications and transmit data in continuous stream, the interface initially aggregates the bytes into larger groups. In our design the data rate is set at 6 Mbps, as a result the physical layer receives data in groups of 24 bits which are stored in a buffer for processing. Secondly the DMA latency cannot exceed 1.6 µs. This is achievable even in a heavy loaded platform where the processor is constantly in demand. However as synchronization is necessary between the DMA and the MAC layer's operation, a buffer of 4 bytes was chosen. This also means that a handshake mechanism is required to allow seamless operation between the layers.

4.2. Java Co-Processor

To enable future capabilities towards distributed computing and IP based networking functions in SB-WSNs, the Java optimized processor (JOP) is integrated as an AHB Master as shown in Figure 8. This new Java co-processor architecture is defined by the memory sharing scheme in place between cores for access to external RAM and is achieved using the AMBA2 bus from ARM [25]. This design operates like a hybrid *multiple instruction stream, multiple data stream* (MIMD) architecture where each processor fetches its own instructions and data. Essentially, it operates thread level parallelism allowing many tasks to be performed simultaneously.

To add JOP as a non-heterogeneous Java based network processor, several issues were resolved:

- JOP Interface: JOP uses the SimpCon bus scheme [27] whilst the LEON3 uses the ARM AMBA2 bus. JOP needs to be added on the shared bus using an interface between the SimpCon and AMBA bus.
- Exceptions: JOP, like any JVM, has exceptions that could cause the processor to stall or exit from operation. These need to be handled to allow for restart of JOP and applications under differing modes and for increased fault tolerance.
- Bootloading: Both the LEON3 and JOP require off chip memory areas, typically in PROMs or FLASH, to hold the software bootloaders. These interfaces must be available to both cores so they can run separately from each other. As JOP avoids dynamic class loading, all required classes must be loaded on startup with known start addresses.

Therefore, integration of the JOP processor has included 1) an AHB Bus Master wrapped for interfacing purposes and connections to the LEON3, 2) an APB slave for communication with the memory controller, and 3) hardware exception handling for automatic recovery as shown in Figure 8.

JOP itself operates 4 pipeline stages: microcode fetch, decode and execute and an additional translation stage bytecode fetch [28]. The core itself uses additional interfaces to find initial start addresses and special pointer addresses. Connections to external components are achieved using the memory core and the extension core. The memory core provides an interface between the main memory and the CPU whilst the extension core provides some extended functionalities including a multiplier unit, control signals for memory and I/O, and a multiplexer for read data to load to the top of the stack.

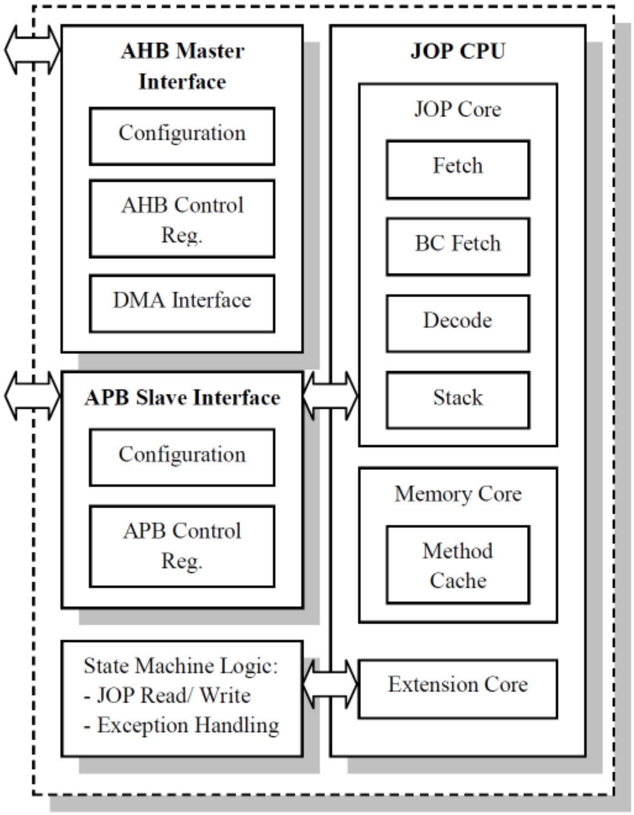

Figure 8. JOP IP Core Wrapper

The original I/O module has been replaced by AMBA interfaces. The AMBA interfaces are an AHB Master interface and an APB Slave Interface where both contain configuration information that is initially sent to the AHB arbiter.

The AHB interface has an additional direct memory access (DMA) interface to perform read and write operations. The APB has some configurable control registers to set start and output addresses of the core as well as feedback for exceptions and debugging DMA signals. The LEON3, with both master and slave functions, is able to request (as a master) and serve (as a slave) to other cores on the AHB bus whereas JOP can only request as a master.

Exception handling has been problematic in fault tolerant systems. Some relevant examples are discussed by H. Hecht [29] including examples of catastrophic failures with an Ariane-5 launcher and the Mars Polar Lander. For a SoC design, there are two types of failures: global failure, where many functional areas of the device are affected requiring a device reset, and a recoverable failure, where processes in hardware and software can cause functional errors in exact areas of the device. To deal with these recoverable errors, there are several main hardware and software exceptions that occur in the Java processor. Hardware exceptions include:

1. Stack Overflow – where the stack becomes full, typically due to a large number of classes
2. Null Pointer – an address which has elements undefined or is out of the memory scope
3. Array Out of Bounds – access to an array element which may not be accessible

Whilst software exceptions include:

1. Network Exceptions – timing constraints not met or unhandled protocol exceptions
2. Application Specific Exceptions

Each of these hardware exceptions typically results in the stalling of the processor and a hard reset is required. The hardware errors are typically due to overloading of the processor or corrupt software whilst the software exceptions occur due to poor network connectivity or programming errors. Therefore, hardware exceptions will all cause an automatic reset and so operationally the processor can be brought back online in the shortest time possible and a register bank is utilized allowing other AHB cores to assess JOP's operational status.

The JOP Java application is first compiled to bytecode, then to microcode, before finally linking with class files. To facilitate the symmetric multiprocessing (SMP) architecture of this design with two heterogeneous processor cores, compilation of each core's application must be stored together in the same image. There are two methods that can be employed to overcome this problem:

1. Embedding the required instructions in a C program and storing them in memory.
2. Compiling each application separately and concatenating using SRecord tools [30] or similar object copy programs at the required addresses.

The LEON3 application has its code (.text segment) typically stored in a PROM at 0x00000000, and data (.data and .bss) in RAM at 0x40000000. At start-up, the .data segment is copied from the PROM to the RAM; linked to start from address 0x0. The data segment for JOP is, by default, linked at 0x4000000 also but can be changed by giving offset arguments; which is the technique used to set JOP's application. JOP's application is aimed at starting at address 0x41000000 and outputting to 0x42000000, away from the LEON3 memory area. These start addresses can be set in a C program by the LEON3 or hard-coded in the JOP IP core wrapper component.

4.3. Agent Middleware

An agent based middleware with instance management is designed for distributed operations in SB-WSNs. Code migration, parallel behaviours and data distribution services are also supported. Both the TCP/IP and the UDP communication protocols are used. The UDP protocol is better suited for 'store-and-forward' communications as a dropped UDP packet, in this case, is preferred to a TCP delayed packet [31]. Use of the protocols depends on the type of data transmission tasks as below:

- 'High Priority Data' tasks use the TCP/IP protocol for reliable and secure point-to-point communication.
- 'Low Priority Data' tasks use the UDP protocol for fast, broadcast/multicasting of small information to groups of satellites employing the publish/subscribe or peer-to-peer communication scheme.

Both types of tasks can take advantage of existing Agent Communication Languages (ACL) [32] for workflow control, acknowledgements and finally support for packet broadcast and multicasting.

There are various agent middleware options available to develop the embedded agent middleware; with the majority using a derivative of JADE [33] or FIPA-OS [34]. But each agent platform has dependencies based on a particular Java revision environment (JRE). For example, JADE can be implemented based on JRE 1.4 or as JADE-LEAP using JRE 1.2. JADE-LEAP can then be configured under J2ME, PersonalJava (or pjava) now superseded by the Connection Devices Configuration (CDC Spec.) [35] and the Mobile Information Device Profile (MIDP) stack which uses the Connection Limited Device Configuration (CLDC Spec.) [36]. FIPA-OS is also considered along with Micro FIPA-OS, targeted for mobile phones.

An in depth comparison of the middleware footprints, RAM usage, and startup time was carried out using a new method probe [37] developed using Eclipse's Probekit from the Test & Performance Tools Platform (TPTP) Project [38] to log RAM measurements at method entry. The key results from

[37] concluded that the JADE-LEAP-pjava is to be used for the final configuration with low RAM and memory footprint as well as a fast startup time. This software configuration offers:

- The CDC stack of standard Java methods usable for networking applications at JRE 1.1.8; either offered by JOP in hardware or in open-source software for emulation.
- The lowest memory consumption when compared to other competing systems.
- Agent functionality through JADE-LEAP with cloning capabilities.

This configuration has been taken forward for development and its footprint reduced to 305 KB using ProGuard [39], an open source Java software tool. ProGuard is employed for shrinking, optimisation, and obfuscation, keeping only the core classes required for the middleware operation and communication. Shrinking analyzes the main application and removes unused classes, fields, and methods. Optimizations include removing debug and logging codes, making classes static and final, and a reduction of variable allocation (mostly coding optimizations). Obfuscation is the replacement of naming in the classes, fields, and methods with simple characters and values. Despite being used to ensure code cannot be reverse-engineered for greater security when the final agent middleware is deployed, it also compacts the code. When compared to previous middleware solutions, this method achieves a reduction of 72% of the existing JADE-LEAP-pjava solution and 64% of a CORBA solution [40], resulting in a very small agent middleware solution is for networked embedded systems.

4.3.1. Middleware Instance Manager

The optimized agent middleware, JADE-LEAP-pjava, needs functionality for autonomous recovery from exceptions. This is achieved using a software wrapper to run JADE-LEAP-pjava as its own manageable thread. An Instance Manager algorithm is developed which manages instances of the JADE-LEAP-pjava agent middleware. As a result the JADE-FT (fault-tolerant) middleware is completed, where agents are considered services accessible in the network.

Software exceptions can often be problematic leading to programming errors, incompatible client (or peer) code and resource failures. Instead of exiting the program and performing a complete hardware reset, controlled exception exit codes are utilised to restart the thread under a safe profile configuration taking advantage of multiple CDC Java profiles. Once started, the middleware operates in nominal conditions. But if JADE-LEAP-pjava crashes due to an unforeseeable exception, the thread is stopped and not the JVM.

In the event of an exception at loading the middleware instance or during normal operations, it is important to know if the exception a) can be handled and b) if it is expected. An example of an expected exception would be if the satellite node knows that is running out of power or drifting away from the network and a previous profile can be found to recover network services as quick as possible using the CDC Profile N = N – 1 loop. An example of an unexpected exception would be if a failure has occurred due to single event upset (SEU) or singe even latchup (SEL). In this case, all satellite nodes return to a safe mode where CDC Profile N = 0. The safe mode has the standard agents services and attempts to find nearby connections from a reset network connection table.

A key area of interest is when an ad-hoc network consisting of mobile nodes performs topology reconfiguration and a new master 'sink' node is assigned. The method probe was used again to find out the overhead of disconnecting and reconnecting middleware instances and performing soft resets of the middleware, as shown in Figure 9, which displays a log of the memory utilisation when a node successfully connects with another node. Correct operation of the middleware is confirmed by testing unexpected connections and disconnections. If a node suddenly errors out, time is needed for the replicated named agents to be removed from the main node lists before reconnection so all nodes connecting to the main node have an additional delay before connecting. From this point, any node can then use relative position/speeds (or other properties) for topology reconfiguration.

Scalability is a key issue here and as the number of networked nodes increases by 1, the memory consumption also increases which is shown in Point 1 of Figure 9. Upon reconfiguration, however, at Point 2, the instance is destroyed and restarted under new conditions, in this case, as a backup node where messaging and control is not as centralised. From Point 2, it is also observed that double the methods are called for one more additional networked middleware instance to be discovered and added.

These three middleware instances are connected using some key classes: the runtime instance, properties assignments, and profile implementations. The profile implementation interface allows the Instance Manager to set a number of key variables as to how to configure the runtime. These variables will determine if the satellite node is configured as the main node (sink), a backup node (if the sink is removed), or a normal peer. The runtime and profile classes then load an agent container and relevant agents based on the chosen profile. This routine is repeated after the proactive reaction time which provides another layer of abstracted control for autonomy and fault-tolerance to the distributed satellite systems software. These methods hold information on the Agent location and registrations at a cost of approximately 200 KB per Agent platform plus an original 600 KB for the first instance.

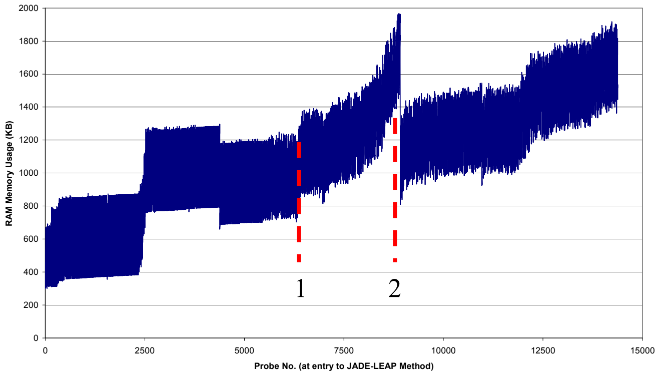

Figure 9. Instance Manager Thread Performing Soft Resets

5. Configurable Intersatellite Communications Module

In view of the dynamic mobility and communications channel characteristics in SB-WSNs, there is a need to develop a configurable communications module (a) to support intersatellite links, (b) to provide longer reconfigurable ISLs, and lastly, (c) to provide relative distance and bearing measurements. This includes very low data rates, and changing operational frequencies.

A prototype ISL communications module is specified aimed at system level testing of distributed processing in the context of a crosslinked constellation mission scenario. Commercial-of-the-shelf (COTS) components will be used for the design. Industrial Scientific and Medical (ISM) frequencies will be employed as operational frequencies. A software defined radio (SDR) based design architecture will be utilised. The ISL module shall satisfy the following key requirements:

- adhere to CubeSat design specifications (PC-104 form factor)
- provide intersatellite communications link at variable data rates and configurable waveforms (adapting to channel characteristics),
- provide ground communications link,
- provide an independent beacon signal generator
- generate localisation information (distance and bearing angles)
- support IEEE 802.11 specifications (IP already developed)

In addition, the same hardware is to act as an integral entity in the SB-WSN test bed, which is under development [2]. A similar test bed is being developed at the Jet Propulsion laboratory and is known as Formation Flying Test Bed [41]..

The functional block diagram is presented in Figure 10. The reconfigurable ISL communications module incorporates S-band (2.4 GHz) as well as 434/144 MHz radio front ends, interfaced to a single reconfigurable modem. A high end AD9861 ADC/DAC [42] is selected for the 2.4 GHz radio front end for a Maxim 2830 radio [43]. A low end high resolution AD7731 ADC/DAC [44] is selected for the 434/144 MHz front end for an Alinco DJC-7E radio [45]. In addition, current sensors and temperature sensors and a 16-bit microcontroller for housekeeping purposes are incorporated in the design. Initial software development is to be carried out on Infineon TriCore TC 1775, 32-bit microcontroller [46].

At present the module design is being validated via prototyping at sub-system level. The RF front-end and the ADC/DAC are being evaluated for a combined system level noise figure and bit error rate under normal room temperatures and extended temperature up to 70 degrees Celsius.

The beacon signal generator is independent from all other sub-systems on the board, but has been designed with data and control interfaces with both the baseband processor and housekeeping controller. It is to provide beacon Morse code encoded at variable rates of 5 wpm to 15 wpm at configurable timing intervals from a default 120 seconds. It is designed to operate as soon as the solar panels generate power and will be the first sub-system to start after confirmation of antenna deployment.

The baseband modem selected is a multi-carrier orthogonal frequency division modulator (OFDM). A Matlab based OFDM transceiver has been implemented for 128 point IFFT, with a 32 point guard band. An advantage of OFDM is that the base bandwidth depends on the sampling frequency of the digital to analogue converter (DAC). Therefore the same modem can operate when sampled at 8 KHz in the audio band (< 4 KHz for the Alinco Radio) and at 20 MHz as a wideband system (< 10 MHz for MAX2830 radio). This would therefore eliminate the need for different modems for different operating basebands. A hardware implementation of the modem is in progress.

6. Conclusions

With the advances in satellite manufacturing, the concept of space based wireless sensor networks is now becoming possible. In particular, a key target environment for SB-WSNs is LEO to investigate space weather phenomena. Future applications utilizing multiple picosatellites in the context of the flower constellation are discussed. A number of orbital and network problems that need to be addressed are outlined in reference to the layers of the OSI networking scheme.

A distributed computing platform for SB-WSNs is proposed that employs several configurable IP cores within a system-on-a-chip design. These include a hardware accelerated WiFi transmitter and a Java co-processor for efficient and adaptable communications. The platform supports an agent based middleware for fault-tolerant networking applications, which enables a hard real-time Java environment when combined with the JOP processor. A new agent middleware configuration with instance management functionality for topology reconfiguration is developed, which is more compact than previous comparable designs. Code migration, parallel behaviours, and data distribution services are also included in the small 305 kB footprint. Under test, it consumes 600 kB RAM with 200 kB for each networked agent middleware instance.

To cater for distributed picosatellite missions, the design of a new configurable intersatellite link communication module is proposed aiming at a low power and low cost implementaion.

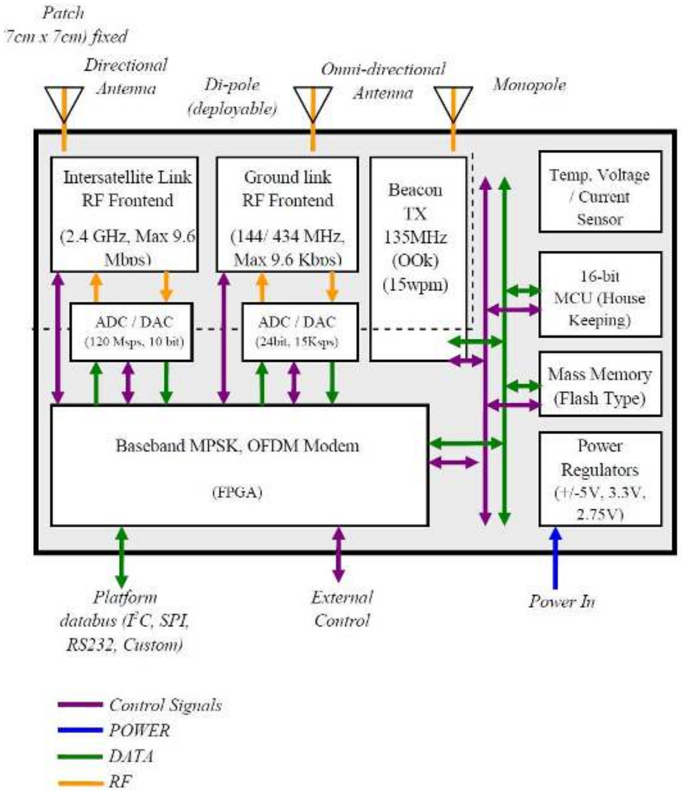
Patch
7cm x 7cm) fixed
Directional Antenna
Di-pole (deployable)
Omni-directional Antenna
Monopole
Intersatellite Link RF Frontend
(2.4 GHz, Max 9.6 Mbps)
Ground link RF Frontend
(144/ 434 MHz, Max 9.6 Kbps)
Beacon TX 135MHz (OOk) (15wpm)
Temp, Voltage / Current Sensor
16-bit MCU (House Keeping)
Mass Memory (Flash Type)
Power Regulators (+/-5V, 3.3V, 2.75V)
ADC / DAC (120 Msps, 10 bit)
ADC / DAC (24bit, 15Ksps)
Baseband MPSK, OFDM Modem
(FPGA)
Platform databus (I²C, SPI, RS232, Custom)
External Control
Power In
Control Signals
POWER
DATA
RF

Figure 10. Intersatellite Communications Module Functional Block Diagram

Acknowledgements

Research funding from the Engineering and Physical Sciences Research Council (EPSRC), UK, under grant EP/C546318/01 is gratefully acknowledged.

References

[1] M. Wilkins, C. Bruccoleri and D. Mortari, “Constellation Design using Flower Constellations”, Paper AAS 04-208 of the 2004 Space Flight Mechanics Meeting Conference, Maui, Hawaii, 9-13 February 2004.

[2] T. Vladimirova, X. Wu and C. P. Bridges, “Development of a Satellite Sensor Network for Future Space Missions”, Proc. of IEEE Aerospace Conference 2008, Big Sky, USA (IEEEAC'08).

[3] T. Vladimirova and D.J. Barnhart. Toward Space Based Wireless Sensor Networks – in “Small Satellites: Past, Present, and Future”, H. Helvajian and S. W. Janson (Eds.), 2008, pp. 595-634, The Aerospace Press, CA, US, 876 p.

[4] N. Iucci, L.I. Dorman, A.E. Levitin, A.V. Belov, E.A. Eroshenko, N.G. Ptitsyna, G. Villoresi, G.V. Chizhenkov, L.I. Gromova, M. Parisi, M.I. Tyasto, V.G. Yanke, “Spacecraft operational anomalies and space weather impact hazards”, Advances in Space Research, Vol. 37, Issue 1, The Moon and Near-Earth Objects, 2006, pp. 184-190.

[5] N.W. Green and J.R. Dennison, “Deep dielectric charging of Spacecraft Polymers by Energetic Protons”, IEEE Transactions on PLASMA Science, Vol. 36, No.5, October 2008.

[6] K.A. Ryden, P. A. Morris, A. Ford, A.D.P. Hands, C.S. Dyer, B. Taylor, C. I. Underwood, D.J. Rodgers, G. Mandorlo, G. Gatti, Hugh D.R. Evans and E. J. Dally, “Observations of internal charging currents in medium Earth Orbit”, IEEE Transactions on PLASMA Science, Vol. 36, No.5, October 2008.

[7] E.L. Afraimovich, V.V. Demyanov, A.B. Ishin, G.Ya. Smolkov, “Powerful solar radio bursts as a global and free tool for testing satellite broadband radio systems, including GPS-GLONASS-GALILEO”, Journal of Atmospheric and Solar-Terrestrial Physics, Volume 70, 2008.

[8] Defence Agency Research Projects Agency, “DARPA Awards Contracts for Fractionates Spacecraft Program”, News Release, 26th February 2008

[9] T. Henderson and D. Mortari, “Uni-Flower: A Novel Proposal for University-Built Nanosatellites in Flower Constellations”, in Proc. of 2006 AAS Space Flight Mechanics Meeting Conference, Tampa, FL, January 22-26, 2006.

[10] C. P. Bridges and T. Vladimirova, “Agent Computing for Distributed Satellite Systems”, in Proc. of 59th International Astronautical Congress 2008, (IAC ’08), Glasgow, UK, October 2008.

[11] Analytical Graphics, Inc. (AGI), Satellite Tool Kit, Website [Online], www.stk.com (last accessed: 08.07.2009)

[12] G. Coulouris, J. Dollimore and T. Kindberg, “Distributed Systems – Concepts and Design” 3rd ed., Section 3.4. Network Principles - Protocols, pp. 76-79

[13] The NASA ASIC Guide: Assuring ASICS for Space, http://parts.jpl.nasa.gov/asic/title.page.html

[14] IPS team, IPS radio and space services a guide to space radiation, IPS radio and space services webpages, http://www.ips.gov.au/Educational/1/2, last visited 2009-10-27.

[15] L. Barclay, “Propagation of Radiowaves”, The Institute of Electrical Engineers, London, UK, 2nd ed, 2003.

[16] K. Sidibeh, T. Vladimirova:” IEEE 802.11 Optimisation Techniques for Inter-Satellite Links in LEO Networks”, Proc. of 8th international conference on Advanced Communication Technology, 2006, ICACT 2006, Vol. 2, 20-22 Feb. 2006, pp. 1177 – 1182.

[17] A.H. Ballard; ”Rosette Constellations Of Earth Satellites”, IEEE Transactions on Aerospace and Electronic Systems, Volume 16, Issue 5, September 1980, pp 656-673.

[18] A.Houyou, R.Holzer, H.de Meer and M.Heidl: “Performance of Transport Layer Protocols in LEO Pico-satellite constellations” Technical report, University of Passau, www.fmi.uni-passau.de/forschung/mip-berichte/MIP-0502.ps.

[19] K. Sidibeh, “Adaptation of the IEEE 802.11 Protocol foe Inter-Satellite Links in LEO Satellite Networks” PhD Thesis, Surrey Space Centre, University of Surrey, UK

[20] X. Wu, T. Vladimirova, and K. Sidibeh, “Signal Routing in a Satellite Sensor Network Using Optimised Algorithms”, Proc. of 2008 IEEE Aerospace Conference, 2008, Big Sky, USA (IEEEAC'08).

[21] CORBA, Website [Online], http://www.corba.org/ (last accessed: 17.06.2009)

[22] Sun Microsystems Inc., Java, Website [Online], http://java.sun.com/ (last accessed: 17.06.2009)

[23] H.Tiggeler, T.Vladimirova, J.Gaisler. Designing a System-on-a-chip for Small Satellite Data Processing and Control, IIE Magazine on Engine ering Technology, vol. 4, N 6, June 2001, pp. 38-42

[24] Gaisler Research, SPARC V8 32-bit Processor LEON3/LEON3-FT CompanionCore Data Sheet, Website [Online]. Available: www.actel.com/ipdocs/leon3_ds.pdf (last accessed: 17.06.2009).

[25] Advanced RISC Machines Ltd. (ARM), *AMBA* Specification Rev 2.0, Specification, 1999.

[26] T. Vladimirova and J. R. Paul. Implementation of an IEEE802.11a Transmitter Module for a Reconfigurable System-on-a-Chip Design - Proceedings of 4th NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009), July 29 – August 1, 2009, San Francisco, California, USA, pp. 305-312

[27] OpenCores, “SimpCon – A Simple SoC Interconnect”, Wesbite,[Online].Available:http://www.opencores.org/projects.cgi/web/simpcon/overview (last accessed:07.07.2009)

[28] M. Schoeberl, “A Java processor architecture for embedded real-time systems”, Journal of Systems Architecture, 54/1-2, pp. 265--286, 2008

[29] H. Hecht, “Requirements for Software Exception Handling”, in Proc. of 2008 IEEE Aerospace Conference (IEEEAC ’08), 1-8 March 2008, Big Sky, Montana, USA, pp. 1-7

[30] SRecord 1.49, Website [Online]. Available: srecord.sourceforge.net (last accessed: 09.07.2009)

[31] L. Wood, C. Peoples, G. Parr, B. Scotney, and A. Moore, “TCP’s protocol radius: the distance where timers prevent communication,”, in Proc. of 3rd International Workshop on Satellite and Space Communications (IWSSC ’07) , Salzburg, Austria, 13-14 September 2007

[32] FIPA Agent Message Transport Service Specification, Website [Online]. Available: www.fipa.org/specs/fipa00067/SC00067F.pdf (last accessed: 17.06.2009)

[33] JADE – Java Agent Development Framework, Front Page, Website [Online], jade.tilab.com/ (last accessed: 17.06.2009)

[34] FIPA-OS at Sourceforge, Website [Online], sourceforge.net/projects/fipa-os/ (last accessed: 17.06.2009)

[35] Java ME Website Connected Configuration (CDC), Website [Online], java.sun.com/products/cdc/ (last accessed: 17.06.2009)

[36] Sun Microsystems Inc., Java ME Website Connected Limited Device Configuration (CLDC), Website [Online], http://java.sun.com/products/cldc/ (last accessed: 17.06.2009)

[37] C. P. Bridges and T. Vladimirova, "Agent Computing Applications in Distributed Satellite Systems", in Proc. of International Symposium for Autonomous Decentralised Systems, Athens, Greece, 22-25 March 2009.

[38] Eclipse Test & Performance Tools Platform Project, Website [Online]. Available: www.eclipse.org/tptp/ (last accessed 05.05.2009)

[39] ProGuard, Version 4.3, Website [Online]. Available: proguard.sourceforge.net/ (last accessed: 02.07.2009)

[40] T. Vladimirova, X. Wu, A. H. Jallad and C. P. Bridges, "Distributed Computing in Reconfigurable Picosatellite Networks", in Proc. of 2nd NASA/ ESA Conference on Adaptive Hardware and Systems, 2007, pp. 682-692

[41] J. Y. Tien, G. H. Purcell, L. R. Amaro, L.E. Young, M. Aung, J. M. Srinivasan, E. D. Archer, A. M. Vozoff, Y. Chong, "Technology Validation of the Autonomous Formation Flying sensor for Precision Formation Flying", Proc. of 2003 IEEE Aerospace Conference, 2003, Vol. 1, March 8-15, 2003.

[42] AD 9861, 10-Bit Mixed-Signal Front-End Processor, Datasheet [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD9861.pdf (last accessed: 01.11.2009).

[43] MAX2830 2.4 GHz to 2.5 GHz 802.11 g/b RF Transceiver with PA and Rx/Tx/Diversity Switch, Webpage [Online]. Available: http://www.maxim-ic.com/quick_view2.cfm/qv_pk/5367 (last accessed 01.11.2009).

[44] AD7731 Low Noise, High Throughput 24-Bit Sigma-Delta ADC, Datasheet [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD7731.pdf (last accessed: 01.11.2009).

[45] Alinco DJC7T/E Specifications, Webpage [Online]. Available: http://www.alinco.com/Products/DJC7/ (last accessed: 01.11.2009).

[46] Infineon Technologies, "TC 1775 Highly Integrated 32-Bit TriCore TM-based Microcontroller for Automotive Applications", Datasheet [Online]. Available: http://www.infineon.com/dgdl/TC1775_pb_V3.pdf (last accessed: 01.11.2009)

BIOGRAPHY

Tanya Vladimirova, MEng, MSc, PhD, CEng, MIET, MEEE, received the M.Sc. degree from the Technical University of Sofia, Bulgaria, the MEng and the Ph.D. degrees from the St. Petersburg Electro-Technical University (LETI), Russia. She is currently a Reader in the Department of Electronic Engineering at the University of Surrey and leads the VLSI Design and Embedded Systems research group at the Surrey Space Centre. Her research interests are in the areas of low-power on-board integrated circuit design, image processing, intelligent embedded systems and space-based wireless sensor networks. She acted as a co-chair of the Military and Aerospace Applications of Programmable Logic Devices (MAPLD) confererence from 2000 to 2006.

Christopher P. Bridges, BEng, PhD in the VLSI Design and Embedded Systems research group at Surrey Space Centre, UK, between 2006-2009 and is now a Researcher at Surrey Space Centre. His research interests are distributed computing, software agents, satellite systems, and multi-core design for FPGAs.

Jean R. Paul, BEng Telecommuncations Engineering (2006), is a PhD student in the VLSI Design and Embedded Systems research group at Surrey Space Centre, UK his research interests include intersatellite communications, cross-layer adaptation in wireless networks, signal processing, FPGA development for embedded systems and computing.

Saad A. Malik, BEng, MSc, is currently a research student at the Surrey Space Centre, University of Surrey, UK. He has three years of design experience in embedded systems development for secure wireless systems and another two years of technical management of core (switch), local loop and Hybrid Fiber Coaxial network from its design to roll out phase. His research interests are in the wireless communications for localization challenges in space based Ad-hoc networks of very small satellites.

Professor Sir Martin N. Sweeting, B.Sc.Hons., PhD (Surrey), FRS, FREng., FIET, FRAeS, FBIS, SMIEEE, SMAIAA, MBIM, MIAA has pioneered the concept of advanced microsatellites utilizing modern commercial-off-the-shelf (COTS) devices for 'affordable access to space.' After completing BSc & PhD degrees at the University of Surrey, in 1985 he formed a spin-off University company (SSTL - Surrey Satellite Technology Ltd) which has designed, built, launched and operates in orbit a total of 34 nano, micro, and mini-satellites - making SSTL the world's leading microsatellite company. As Chief Executive of SSTL, he has been responsible for the leadership and management of the Company which by 2006 has grown to 210 commercial staff and achieved a total export sales of over £110M. Sir Martin is also Director of the Surrey Space Centre, leading a team of 80 faculty and doctoral researchers investigating advanced small satellite concepts and techniques. Sir Martin was knighted by HM The Queen in the 2002 British New Year Honours for services to the small satellite industry.