BioNet Middleware and Software Framework in Support of Space Operations

Kevin K. Gifford*, Shea Williams†, Lenny Maiorani‡, Brian Marshall§ and Marek Sotola**

BioServe Space Technologies, University of Colorado, Boulder, CO, 80309 USA

I. Introduction

BioNet is a software framework that provides plug-and-play operations for hardware and software used in the computerized control of automated systems. BioNet is used to help independent software developers (programmers) build and utilize a distributed network data acquisition system. The designers of software frameworks aim to facilitate software development by allowing developers and programmers to devote their time to meeting project requirements rather than dealing with standard low-level details of providing a working system, thereby reducing overall development time.

The BioNet software framework is implemented as a middleware software product. Middleware software provides services to assist application developers in composing new software programs. The BioNet middleware is composed of two entities: (1) The BioNet software framework; and (2) The BioNet developers kit (or “DevKit”).

BioNet’s unique advantage is that it enables all digital devices and networks “to talk to each other” enabling interoperability, reducing costs and eliminating vendor lock-in. This advantage would be useful in a variety of commercial applications.

The key function of BioNet is data collection, storage and management from a distributed network of wired and wireless sensing and control devices. The BioNet software framework provides data interoperability by normalizing the data from the different data-producing sensors and control devices. The software framework can store, transport and display voice, video and data in a single unified data management system.

BioNet was expressly designed to facilitate the incorporation of disparate data producers into a single, unified data management system with a standardized application interface that enables a multi-vendor solution. University of Colorado researchers developed the BioNet software framework in conjunction with NASA. NASA funded research to develop a software framework to standardize software development among its contractors, grantees and own internal software developers. With this standard software framework, NASA can direct a multi-institution software development; NASA can direct its contract, grant and internal software developers to use the BioNet software framework to develop software to collect data from a variety of digital devices, networks, sensors, and other automated systems.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of relevant related work; Section 3 provides an architectural overview of the BioNet framework; Section 4 overviews several BioNet space applications currently in use and under development; Section 5 provides a summary of commercial application markets for which the benefits of BioNet, developed with NASA research funding, are applicable; Section 6 identifies important future work; and Section 7 provides a paper summary with conclusions.

II. Related Work

The concept of middleware, which can be defined as software that manages connections between data produces...
and data consumers, evolved from the topical area of distributed systems. Historically, middleware for distributed systems has emphasized homogeneous computing platforms, i.e., all computing devices were identical or all of the same general device classification. Two of the fundamental problems to overcome initially in this domain were a scalable naming schema and the issue of time synchronization, especially with respect to total ordering of events [1][2]. The original application drivers for distributed systems were file sharing and redundancy/replication. Common middleware architectures in this regard include CORBA, DCOM, .NET, and the JAVA RMI.

Recently, middleware-based distributed systems have received interest primarily fueled by advances in telecommunications systems and the still-growing mobile cellular phone marketplace. In the cell phone market, as in most distributed grid or cluster computing markets, the deployments are characterized by communication between a set of homogeneous devices (e.g., cell phones, or PC-based grid infrastructures). Heterogeneity has been involved in these systems only in terms of different wired network protocols (e.g., Ethernet and Token Ring); support of both wired and wireless (Ethernet LAN and WLAN) installations; along with delivering heterogeneous media content (such as text, voice and video) to a set of devices. Recent middleware and wireless networking research has concentrated on design issues [3][4], wireless device roaming [5], data aggregation [6], and publication/subscription architectural analysis for these types of systems [7], high availability, and certainly on distributed system security. However, the BioNet middleware system, that by design explicitly incorporates a disparate set of heterogeneous wireless devices, is a novel concept. NASA’s need to provide advanced environmental and physiological monitoring, which is required for manned exploration missions, is a unique application driver in this problem domain.

Two well-known examples of software frameworks that are analogous to BioNet are the OMG DDS framework implemented in OpenDDS [9] and the Microsoft .NET framework [10]. Two additional common frameworks that are widely deployed in commercial market sectors [11] include the open-source Google Web Toolkit [12], GWT, and the Java Native Interface, JNI, framework that is an open specification [13].

III. BioNet Framework Architecture Overview

Software frameworks have distinguishing features that separate them from libraries or normal user applications; the BioNet framework adheres to all of the below characteristics [8]:

- **Inversion of control** - In a framework, unlike in libraries or normal user applications, the overall program's flow of control is not dictated by the caller, but by the framework.
- **Default behavior** - A framework has a default behavior. This default behavior must actually be some useful behavior and not a series of no-ops.
- **Extensibility** - A framework can be extended by the user usually by selective overriding or specialized by user code providing specific functionality.
- **Non-modifiable framework code** - The framework code, in general, is not allowed to be modified. Users can extend the framework, but not modify its code.

The NASA Command, Control, Communications and Information (C3I) Interoperability Specification (NASA CxP 70022, Vols. 1 - 7) has been proposed as an architectural approach to enable the “complex orchestration of major new human and robotic vehicles and ground systems which are developed by many different sources” [14]. BioNet traces much of its inspiration to the NASA C3I Specification requirements. With BioNet, the focus is on enabling a “system of systems” and the ability to evolve while meeting spaceflight software certification constraints.

While several commercial software frameworks are available, none of these were suitable for NASA purposes because of (1) the demand for high-power computational processors required by many frameworks was undesirable; and (2) importantly, NASA desired an open standard specification with potentially an open-source software reference implementation. Such an open implementation could run on common platforms including Apple, Linux, and Microsoft Windows.

**BioNet technical definition:** BioNet is a multi-platform, peer-to-peer, publish/subscribe, data-centric, middleware framework. In more detail:
**Multi-platform:** BioNet is a software development framework for the Linux and Microsoft Windows computer operating system platforms.

**Peer-to-peer:** BioNet communicates with other BioNet entities directly over network sockets, avoiding a centralized messaging broker, or server. BioNet locates other entities (peers) with mDNS-SD [15], which is a collection of IETF [16] standards.

**Publish/subscribe:** BioNet utilizes the widely used publish/subscribe network traffic model where updates are sent when available, as opposed to polling the data source continuously.

**Data-Centric Middleware:** BioNet is designed for use with C-language data structures, as opposed to generic messaging middleware systems, which are designed to send byte arrays.

**Framework:** BioNet doesn't actually do anything by itself. BioNet is a middleware framework that can be used by an independent implementer, and customized for the particular project at hand.

The BioNet peer-to-peer architecture is shown in Figure 1. A BioNet network is made up of a set of peers. Peers can join and leave the BioNet network at any time. In Figure 1 there are two kinds of peers: "Hardware Abstractors" (HABs) and "Clients".

HABs interface with specific sensors and effectors and export (publish) these devices and their data to the BioNet network. HABs are publishers of sensor data and consumers of actuator settings.

Clients get data from HABs and process it, and provide settings for actuators to HABs. Clients are subscribers of sensor data and providers of actuator settings. Clients choose what data they want to subscribe to. Together the HABs and clients perform information exchange utilizing efficient publication/subscription (pub/sub) communications.

![Figure 1 – The BioNet peer-to-peer architecture (from [14]).](image-url)
The BioNet Developer Kit is provided as open source software to independent developers so that any vendor device can be integrated into the system. The “Dev Kit” provides a C source code module that uses standard C-language libraries that serve as a detailed example for any developer. In addition to C-language bindings, Perl and Python bindings are also available. Software developers utilize the DevKit to interface to system hardware and to compose BioNet application clients to ingest BioNet data.

The BioNet Development Kit works by provides example programs and template source code for a software developer. Network communications along with standard data bus messaging are provided via the BioNet framework. By starting with a framework, the software designer does not need to understand the different protocols required for network and device communication. Instead a developer can begin to develop specific software capabilities as required for a particular data acquisition project. Using a software framework reduces overall software development time and aids in ensuring conceptual integrity of a distributed data acquisition system, leading to improved system reliability and increased system robustness to failure.

**BioNet Data Manager:** BioNet is a "soft real-time" system that aims to disseminate up-to-date information to interested parties. The BioNet Data Manager (BDM) is a component of BioNet that records and provides a historical record of the data that flows over BioNet.

The BDM joins the BioNet network as a Client. It subscribes to all BioNet data using the regular BioNet Client interface, and records the data it receives. Recorded data is made available to the network using a pub/sub protocol suite similar to the BioNet protocols. In addition, the BDM communicates with other BDM processes utilizing Disruption Tolerant Networking, DTN, to relay the data and make it available to remote BioNet peers.

**Disruption (and/or Delay) Tolerant Networking:** Delay tolerant networking is an enabling technology for the Interplanetary Internet envisioned to provide communications for NASA’s Exploration Mission activities. Long-haul, or interplanetary, communications links (e.g., Earth-to-Moon and Earth-to-Mars links) must contend with large transmission latencies, potentially high bit-error rates, and asymmetric data rates on the communication channels. Additionally, the availability of the communications links (connectivity) is variable due to periodic orbits of communications relay satellites and availability of terrestrial assets, such as the Deep Space Network (DSN), to participate in the multi-hop communications link.

![Figure 2](image-url) – A general network diagram, depicting four nodes used for Earth-to-Lunar habitat communications, with four peer hosts that employ delay tolerant networking over potentially intermittent links. The fundamental concept is that DTN hosts (nodes) employ local non-volatile disk storage (instead of only random access memory, RAM) to buffer messages and files until acknowledgement of delivery to a downstream entity is successfully received.
Delay, or disruptive, tolerant networks make use of *store-and-forward* techniques within the network in order to compensate for intermittent link connectivity. In Delay Tolerant Networking [17] the fundamental concept is an architecture based on Internet-independent middleware where protocols at all layers are used that best suit the operation within each environment, with a new overlay network protocol (the “bundle protocol”) inserted between the applications and the locally optimized communications stacks (see Figure 2). Many applications can benefit from the reliable delivery of messages in a “disconnected” network.

The Internet, in contrast, is a “connected” network where Internet protocols, most notably TCP/IP, are dependent upon (low) latencies of approximately milliseconds. This low latency, coupled with low bit error rates (BER), allows TCP to reliably transmit and receive acknowledgement for messages traversing the terrestrial Internet.

One of the best examples of high latency, high BER links, with intermittent connectivity is that of space communications. One-way trip times, at the speed of light, from the Earth to the Moon incurs a delay of 1.7 seconds; while one-way trip times to Mars incur a minimum delay of 8 minutes. The problem of latency for interplanetary links is exasperated with increased BER due to solar radiation. In addition, the celestial bodies are in constant motion, which can occlude the required line-of-sight between transmit and receive antennas, resulting in links that at best are only intermittently connected. Intermittent link connectivity is commonplace terrestrially as well. One example is the plethora of battery-powered mobile communications devices that (1) go in and out of communication range to wired service interface points (cellular base stations, BSs, and WLAN access points, APs); and (2) are turned on and off at the user’s discretion. Military applications in the DTN arena are substantial, allowing the retrieval of critical information in mobile battlefield scenarios using only intermittently connected network communications. For these types of applications, the delay tolerant protocol should transit data segments across multiple-hop networks that consist of differing regional networks based on environmental network parameters (latency, loss, BER). This essentially implies that data from low-latency networks for which TCP may be suitable must also be forward across the long-haul interplanetary link. DTN achieves message reliability via employing “custody transfer”. The concept of custody transfer, where responsibility of some data segment (bundle or bundle fragment), migrates with the data segment as it progresses across a series of network hops is a fundamental strategy such that reliable delivery is accomplished on a hop-by-hop basis instead of an end-to-end basis which is impractical over high latency links.

Together, BioNet and DTN improve data timeliness associated with robotic and human-tended missions, assisting NASA in reducing risk, reducing cost, increasing crew safety, improving operational awareness, and improving science return.

### IV. BioNet Applications in Support of Space Activities

Potential near-term NASA applications of the BioNet architecture include vehicle health monitoring of the Shuttle and other heavy lift launch vehicles, structural and environmental monitoring in and around the International Space Station (ISS), crew medical monitoring, and as a command and communications component for Space Exploration. As part of the Exploration Directorate, proximity networks could easily be deployed on the Moon and Mars, with inherent interoperability and coexistence capabilities even when provided by multiple developers. Such networks could include imagery collection, crew communication, remote sensing nodes for scientific applications, robotic command and control, and environmental and safety monitoring.

BioNet is utilized for day-to-day activities in support of 2 NASA-related R&D activities:

1) Life science experiment support within payloads onboard the International Space Station (ISS); and
2) As an automated inventory management system under development at NASA Johnson Space Center.

BioNet software modules are utilized to control and communicate with the different experiment hardware in a Commercial Generic Bioprocessing Apparatus (CGBA) science payload (see Figure 3). BioNet is used to provide standard automated system capabilities including experiment initiation and termination, thermal control for life science specimens, experiment image acquisition and downlink of science telemetry from the payload to a ground-based payload operations control center.
BioNet is currently deployed within the NASA Johnson Space Center terrestrial-based Lunar Habitat Wireless Testbed, LHWT [27]. BioNet is utilized for the integration of various inventory management hardware, including handheld and portal-based radio frequency identification, RFID, systems, “smart” shelves, and the “smart” receptacles for RFID asset tracking, as shown in Figures 5.

BioNet was ported to the Apple iPod/iPhone for the NASA LHWT project, thereby providing a mobile handheld command and control device (see Figure 4).

Table 1 on the following page summarizes features and benefits of the BioNet software framework and communications architecture. Note that while IT security design of BioNet is not discussed in this paper, BioNet does provide integrated data privacy, data integrity and message authentication services.
Table 1 – Benefits and features of the BioNet software framework for space exploration.

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data interoperability</td>
<td>BioNet integrates data from disparate data-producing sensors, controllers. Voice, video and data can be stored, transported and display by a single unified control and communications system.</td>
</tr>
<tr>
<td>Multi-vendor solution set</td>
<td>Multi-vendor hardware (instrumentation, controllers, radios, etc.) can be considered for the engineering solution. Non-proprietary (open) as well as proprietary solutions can both be integrated within the BioNet enterprise software framework.</td>
</tr>
<tr>
<td>Independent development</td>
<td>The BioNet framework and Development Kit (DevKit) enables system software development by multiple independent software developers of: (1) software to interface with hardware devices and (2) client applications that can make use of any piece of system data for decision-making.</td>
</tr>
<tr>
<td>Mobility</td>
<td>BioNet is explicitly designed for seamless roaming, enabling untethered mobility of sensors, instrumentation equipment, communications devices, and personnel for maximum productivity.</td>
</tr>
<tr>
<td>Standards-based interoperable network communications</td>
<td>Allow for interoperable use of differing radio (RF) technologies along with traditional wired data transmission for maximum flexibility [RFID, IEEE 1902.1 (Rubee), 802.11 (Wi-Fi), 802.15.1 (Bluetooth), 802.15.4 (ZigBee), 802.16 (Wi-Max), 802.22 (WRAN), IEEE 802.3, USB and serial].</td>
</tr>
<tr>
<td>Scalability</td>
<td>The BioNet peer-to-peer distributed architecture enables highly scalable systems from tens of devices to thousands of data-producers enabling true 'systems of systems' scalability.</td>
</tr>
<tr>
<td>Network robustness and reliability</td>
<td>The BioNet peer-to-peer architecture automatically reconfigures to enable entire subnetworks to enter or leave the communications and control network. The peer-to-peer architecture keeps operating in the face of network partitions and is self-healing to allow networks to recombine.</td>
</tr>
<tr>
<td>Platform independence</td>
<td>BioNet adheres to open-architecture and provides standards-based solutions for maximum interoperability.</td>
</tr>
<tr>
<td>Ease of application composition</td>
<td>BioNet provides helper libraries and services to provide unified interface to device and service discovery, time synchronization, security, compression, etc. The middleware acts as the &quot;service broker&quot; offering a set of services with defined interfaces for ease of implementation.</td>
</tr>
<tr>
<td>Eases burden of flight software certification</td>
<td>The BioNet framework is designed to provide standardized services for application developers, easing the software development process and ensuring conceptual integrity of the system.</td>
</tr>
<tr>
<td>Unified enterprise data management system</td>
<td>BioNet integrates future and legacy heterogeneous devices into a common system based on an open architecture concept that supports widely recognized IEEE standards, as well as supporting vendor-specific (proprietary) protocols for unique applications.</td>
</tr>
<tr>
<td>Automated data replication and back-up</td>
<td>Provide automated database synchronization mechanisms that synchronize data stores as BioNet nodes move dynamically in and out of the network.</td>
</tr>
<tr>
<td>Consistent operational interface</td>
<td>Abstracting networks and hardware for the application developer simplifies composition of common command, control and monitoring tools. This directly results in crew time minimization.</td>
</tr>
<tr>
<td>Integrated network security</td>
<td>BioNet provides integrated security services for data privacy, data integrity, authentication and authorization.</td>
</tr>
<tr>
<td>Delay tolerant communications</td>
<td>BioNet provides reliable message transfer across high-latency, intermittently-connected links enabling networked interplanetary communications.</td>
</tr>
<tr>
<td>Engineering data analysis</td>
<td>BioNet provides a historical database for parameter trending evaluation with easily customizable user displays for optimal graphical visualization.</td>
</tr>
<tr>
<td>Easy support of add-on or retro-fit activities</td>
<td>BioNet facilitates the late or retro-fit addition of additional sensors, controllers, and communications equipment to add additional required functionality as test and verification activities impose new requirements or new capabilities are deemed necessary.</td>
</tr>
<tr>
<td>Endpoint re-programmability</td>
<td>BioNet enables endpoint devices (sensors, radios, etc.) to be selectively chosen and dynamically reprogrammed to upgrade functionality while maintaining nominal operation of the control system.</td>
</tr>
</tbody>
</table>


V. Commercial Applications

Table 2, below, summarizes the current commercially applicable prototyping activities of the BioNet software framework.

<table>
<thead>
<tr>
<th>Commercial Market Sector</th>
<th>BioNet Software Framework R&amp;D, Prototyping, and Project Demonstration Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Management</td>
<td>BioNet is used to provide automated inventory management systems utilizing RFID-tagged items, portals, and hand-held readers. RFID hardware utilized is from Alien Technology [18], Impinj [19], and ACC Systems [20] commercial vendors.</td>
</tr>
<tr>
<td>Computer health monitoring</td>
<td>BioNet is deployed on desktop computers where it is used to monitor the health of the computer that it is deployed upon (CPU performance, disk capacity and health, load monitoring, etc.).</td>
</tr>
<tr>
<td>Cell phone / PDA</td>
<td>BioNet has been ported to operate on Apple iPods and iPhones for mobile handheld monitoring devices. A BioNet port to the Google Android cell phone is currently under investigation.</td>
</tr>
<tr>
<td>Digital voice and video communications</td>
<td>BioNet is used to provide connection management for peer-to-peer digital voice and video transit.</td>
</tr>
<tr>
<td>Building Automation and Home Automation</td>
<td>BioNet is used in building and home automation with power utilization and optimization as the leading application.</td>
</tr>
<tr>
<td>Medical informatics / patient monitoring</td>
<td>BioNet is used to provide unified communications to medical monitoring devices including the CPOD [21], for Bedrest studies [22], and for ECG, blood pressure monitoring and digital stethoscopes [23].</td>
</tr>
<tr>
<td>Environmental monitoring</td>
<td>BioNet has provided integration of hardware for environmental monitoring systems. Specific commercial vendor hardware integrated includes the Crossbow wireless sensor network motes [24], the Industrial Scientific Corp. Compound Specific Analyzer for Combustion Products (CSA-CP) for atmospheric monitoring [25], and the Invocon MicroWIS CO2 wireless sensor [26].</td>
</tr>
</tbody>
</table>

Table 2 – BioNet software framework current R&D uses.

VI. Future Work

BioNet is an ongoing iterative research and development project and there are several areas that remain to be explored and/or implemented.

Several features are planned to be implemented. These include: adding unit and media-type attributes to resource data points; adding support for commanding over DTN links; recording online/offline time stamps in the BDM, and enhancing streams to support multicast and multiple CODECS.

The BDM interface is currently being enhanced to operate in a peer-to-peer publish and subscribe mode. This will enhance BDM monitoring performance, and provide a more unified view of BioNet data.

Security will be enhanced to include per-resource authorization tables, to allow data with differing sensitivities to coexist in the BioNet network. It is expected that PKI will be used to distribute signed resource name pattern ACLs to clients that can be verified off-line by HABs. Combined with existing PKI authentication and encryption, this should allow for authorized peer-to-peer communications in network segments that are disconnected from the authorization authority.

Based on anecdotal evidence, the performance of the BioNet system is adequate. However a true performance characterization will be done in terms of network utilization, and host resources.

Currently creating classes of HABs that perform similar functions, such as “thermometers” or “accelerometers” is left as an exercise to the implementor. This process is adequately accomplished in small systems by using descriptive HAB and Node names, such as “thermometer”. For large systems and long term deployments, it is expected this method will quickly become intractable. A solution may be included in BioNet to define classes of
devices with required and optional resources that are enforced by the BioNet libraries to ensure compatibility with specialized clients, and automated data monitors.

Quality of Service (QoS) attributes are expected to become a necessity to facilitate graceful degradation of BioNet traffic in situations where network links are overloaded. Since some of the data subscribed to over BioNet may not be necessary, there should be a mechanism to flag it as lower priority so that it is sent only when the bandwidth is available for it. A similar feature that allows data points to be merged to create a lower resolution data set may also be considered.

A port of the BioNet library to alternate platforms is on the list of future works for platforms such as RTEMS and VxWorks. Currently Linux (Ubuntu, Debian, Red Hat, Gentoo), iPhoneOS, and MacOSX are supported. A BioNet port for Windows and Windows CE is in progress.

VII. Summary and Conclusions

This paper has provided an overview of the BioNet software framework whose focus is to provide a complete set of standardized services to application programmers in order to simplify, standardize and improve the efficiency of the software development process. The BioNet Developer Kit is provided as open-source software to independent developers so that any vendor device can be integrated into the data management system. The ‘DevKit’ enables development of: (1) software to interface with hardware devices and (2) client applications that can make use of any piece of system data for decision-making. The BioNet middleware provides fundamental device and service discovery of network peers. An enabling technology for complex distributed systems, the BioNet middleware acts as the “service provider” offering a set of services with defined interfaces for ease of implementation. Abstracting networks and hardware for software developers simplifies composition of common application software services.

The BioNet software framework and data management system addresses the fundamental problem of command, control, and communications systems interoperability. Interoperability of Space Exploration systems is necessary to improve reliability, reduce complexity, increase software and hardware usability, and enable multi-developer / multi-agency support [29][30].

BioNet reduces cost and risk by directly enabling interoperability (a common goal of middleware) and reduces overall system complexity by providing common services (e.g., security, quality of service, device discovery, etc.) in middleware – coded once and reused. This design directly minimizes flight software certification costs that often are the primary software development cost for NASA. BioNet, as a middleware, can either replace or exist in parallel with current data acquisition and communication systems, easing the transition from a significant installation base of legacy systems.

BioNet is explicitly designed for seamless roaming, enabling untethered mobility of sensors, instrumentation equipment, communications devices, and personnel to maximize productivity. BioNet employs a peer-to-peer distributed architecture that enables highly scalable systems from tens of devices to thousands of data-producers, thus enabling true ‘systems of systems’ scalability. The peer-to-peer architecture automatically reconfigures to facilitate entire sub-networks to dynamically enter or leave the distributed communications network [28]. BioNet continues nominal operations in the face of network partitions and is self-healing to allow networks to recombine. The data generated by multiple hardware devices and software programs is stored in a standard database to provide historical data for parameter trending and evaluation. The system facilitates the retrofit add-on of additional sensors, controllers, and communications equipment to supplement baseline functionality.

BioNet provides integrated Disruption Tolerant Networking, DTN, communications. For space-based operations, DTN is necessary to enable network communications utilizing multiple communication assets and network paths for improved robustness of the communication network. BioNet-DTN enables increased timeliness of data return from operating space assets. By improving data timeliness DTN communications assist in reducing risk, reducing cost, increasing crew safety, improving operational awareness, and improving science return, all of which lead to an increased return on investment for the agency.
For non-space or terrestrial applications, BioNet is currently being studied in research environments for a number of applications including: process control, data acquisition, building and home automation and the health sector.

Acknowledgments

The authors would like to acknowledge the support of the NASA DTN Readiness Group, Dr. Vint Cerf, and the helpful support of the NASA ION Working Group. This work was performed under NASA contract NNJ05HE10G.

References

22. Bedrest paper reference