



**UNIVERSITY**  
*of*  
**GLASGOW**

St Ville, L. and Dickman, P. (2003) Garnet: a middleware architecture for distributing data streams originating in wireless sensor networks. In, *23rd International Conference on Distributed Computing Systems Workshops, 19-23 May 2003*, pages pp. 235-240, Providence, Rhode Island.

<http://eprints.gla.ac.uk/3492/>

# Garnet: A Middleware Architecture for Distributing Data Streams Originating in Wireless Sensor Networks

Lyndell St. Ville and Peter Dickman  
Department of Computing Science, University of Glasgow, Glasgow, G12 8QQ, UK  
{lsv,pd}@dcs.gla.ac.uk

## Abstract

*We present an architectural framework, Garnet, which provides a data stream centric abstraction to encourage the manipulation and exploitation of data generated in sensor networks. By providing middleware services to allow mutually-unaware applications to manipulate sensor behaviour, a scalable, extensible platform is provided.*

*We focus on sensor networks with transmit and receive capabilities as this combination poses greater challenges for managing and distributing sensed data. Our approach allows simple and sophisticated sensors to coexist, and allows data consumers to be mutually unaware of each other. This also promotes the use of middleware services to mediate among consumers with potentially conflicting demands for shared data.*

*Garnet has been implemented in Java, and we report on our progress to date and outline some likely scenarios where the use of our distributed architecture and accompanying middleware support enhances the task of sharing data in sensor network environments.*

## 1. Introduction

Advances in miniaturisation and low cost, low power chip design [1, 2, 4, 9, 16] are enabling factors in the deployment of sensing networks which comprise small, wireless, low power, unattended devices. By harnessing the wireless communication capability of such devices, low cost Wireless Sensing Networks (WSNs) are formed.

Such networks are advantageous in scenarios such as environmental monitoring and military reconnaissance as they offer cost-effective, dense and widespread coverage with high signal-to-noise ratios [3].

WSNs are a particular example of real time systems, since they are geared towards the timely recognition and detection of objects of interest. In this paper, we define real-time to be context dependent because WSNs are typically assembled with knowledge of the specific requirements of the monitoring task. The time constraints are driven by the

environmental context in which the sensor network operates and the targets being monitored.

As WSNs become more popular, they can be expected to attract the attention of application developers who wish to access the vast source of data which they represent. Opening such systems to the wider software engineering community requires system-level support for the distribution and management of the data streams being generated. The presence of a middleware infrastructure would be helpful in streamlining access to such data and in reducing the potential for duplication of effort.

To maximise the benefit of collecting data in such real-time environments, the data distribution aspect of such systems should ideally support mechanisms which facilitate the delivery of the data such as:

- Abstractions for node identity and location.
- Mechanisms for relaying control messages to support dynamic changes to the system configuration and behaviour.
- Simple, flexible and secure mechanisms for accessing the data.
- Low performance overhead, scalable design and mechanisms to support quality of service.

To address these issues, we present Garnet, a layered, distributed, middleware architecture with an emphasis on managing data streams as an abstraction. Garnet permits mutually unaware consumers to undertake dynamic control of the sensors and influence the data delivery process, which is desirable since application-level knowledge can be used to improve the overall operation of the network. A policy-driven infrastructure arises, where consumers harness the data handling services provided to effect their operations.

Our Java-based proof-of-concept implementation supports up to 16.7M sensors, 256 internal-streams/sensor, 64K sequence counts and payloads of 64K bytes.

Our initial work has highlighted interesting problems in the generic management of data-streams such as describing sensor constraints, management of conflicting control

requests, and the application of predictive resource management strategies based on state-change information. Our architecture provides hooks through which more sophisticated policies in these areas can be provided and exploited. The emphasis in this paper, however, is on the underlying enabling mechanisms, not the policies themselves, and only simple, straightforward policies are assumed.

In Section 2, we present our motivation and design objectives. In Section 3, we outline the overall assumptions. We present our architecture in Section 4 and describe its major components and data structure.

In Section 5, we elaborate on the rationale behind our key design choices. A summary of the novel aspects of our work is presented and discussed in Section 6. We compare our approach to existing work in Section 7, and in Section 8, we identify the enhancements which we envisage may be introduced in later incarnations of Garnet. We conclude in Section 9 with a summary of our work.

## 2. Motivation and Design

We approach the issue of sharing and manipulating sensor data streams from a general perspective where a return path is anticipated for receiving control messages.

In the literature, send-receive capability of sensor nodes is highlighted only as far as inter-node communication facilitates sensor-level coordination for localisation [8], data fusion and range estimation [6], data diffusion [13], or power-control and protocol adaptation [10] purposes.

We believe the focus on transmit-only sensor networks will eventually give way to a broader recognition of, and interest in, send-receive networks. Our design choices reflect this future need where sensor-network specific middleware will be required to facilitate broad access to and manipulation of already-deployed networks.

As a key design feature, consumer processes are mutually unaware, which may lead to conflicting interaction with the sensor field. The middleware support provided to manage this interaction is discussed further in Section 4.

Our approach contrasts with others such as [14, 15], which adopt a database-centric view of querying and sharing sensor data, and where the extent of application-level involvement is restricted to issuing queries on the data. Such approaches lack the flexibility required to support a suitable abstraction for direct programmer manipulation. Also, the restricted view of the sensed data only allows specific combinations of queries to be answered. A more flexible approach is needed to facilitate interaction with the network of sensors in a realistic manner.

By emphasising the importance and flexibility of the data streams, we facilitate ease of separation of the data from the object of interest and remove the emphasis from the physical artefact being monitored. To access location information we provide a location service which treats location data as any other data stream since, depending on the context, lo-

cation information may be regarded as sensitive and should be protected by additional security mechanisms.

## 3. Overview and Assumptions

Due to paper length restrictions, extensive description will be reserved for the more interesting and unique aspects of our work. Full details are available from [17].

In our model, mobile sensors transmit data over an unreliable wireless medium to a fixed network infrastructure.

At the fixed network, the data is consumed by applications which use typical advertising, discovery, registration, authentication and publish/subscribe mechanisms to identify, subscribe to, and receive data streams of interest.

Consumer processes may attempt to influence the future contents of the originating data streams by invoking middleware services. Unless otherwise indicated, communication is based on asynchronous message exchange.

Service-level parallelism and replication are not explicitly featured, although their existence shall be presumed for efficiency, data-integrity, and fault-tolerance.

Additionally, the middleware services are all presented as logically separate and distinct entities, although some combinations of functions co-exist for practical reasons.

## 4. System Architecture

Figure 1 shows a simplified diagram of the services defined by Garnet for processing the sensor data streams. The middleware support provided by Garnet is presented in Section 4.2, where its internal composition and interactions are highlighted.

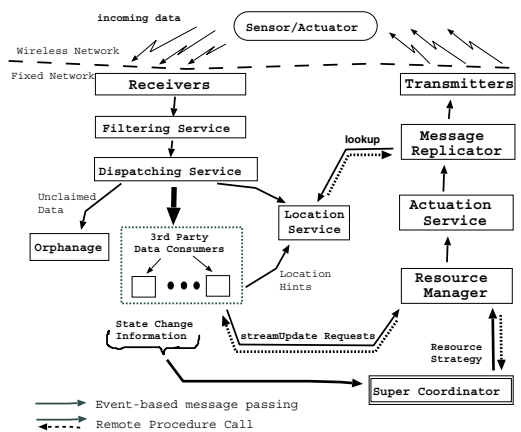


Figure 1. Garnet Architecture

### 4.1. Overview

Figure 1 represents the mobile sensors which wirelessly transmit data to, and may receive data from, a fixed network

infrastructure. The arriving data undergoes a filtering process before being dispatched to consumer processes which may choose to influence the future actions of the sensors. Consumer processes send messages along a return actuation path made available for control messages to be routed to the target sensor in the wireless network. Additional services process the received data and may interact with the consumer processes to affect the overall policy and strategies being employed by the middleware.

## 4.2. Components

### Sensors/Actuators

Sensors transmit data messages which are picked-up by an array of receivers. Sensors are expected to occasionally roam outside the reception zone, which may cause data messages to be lost.

### Receivers

These are arranged such that their effective receiving areas may overlap. Such coverage improves data reception but causes potential duplication of data messages.

### Filtering Service, Dispatching Service

The Filtering Service reconstructs the data streams by eliminating duplicate data messages. Filtered data is then forwarded to the Dispatching Service for delivery to subscribed consumer processes.

### Consumer Processes and Services

Consumer processes use a publish/subscribe mechanism to access data streams, which permits un-configured data streams to be detected.

- System-defined consumer services

The Orphanage is a default consumer process which receives un-configured data. There, data messages are analysed and potentially stored.

The Location Service receives location information which is inferred by the Receivers. This data is mainly used to target location areas when transmitting control messages to the sensor field.

Consumers processing data from location-aware sensors may supply location hints to the location service.

- Multi-level consumers

Consumer processes may generate further derived data streams by performing additional processing on received data. By supporting multi-level data consumption where each layer offers increasingly enhanced services to successive levels, an arbitrarily rich application infrastructure can be assembled.

### Super Coordinator

Suitably sophisticated consumer processes may forward state-change details to the Super Coordinator, which eventually amasses a global view of these consumers. In response to (or in anticipation of) global consumer states, the Super Coordinator may invoke policy changes in the strategy used by the Resource Manager.

### Resource Manager, Actuation Service

A pathway exists for consumer processes to transmit control messages to sensors in a location-neutral manner. First, approval is sought from the Resource Manager which exercises control over the permissible actions which a set of consumers may request. The Actuation Service next processes the request with timestamps, and checksums, before forwarding to the message replicator.

### Message Replicator, Transmitters

The Message Replicator determines the expected location area of the target sensor. Based on the location area, the appropriate set of Transmitters broadcast the request, whereupon it may be received by the sensor node.

## 4.3. Data Messages

A Garnet data message consists of a header followed by a variable length payload as illustrated in Figure 2. A data-stream is formed by a sequence of these messages.

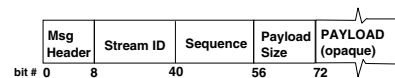


Figure 2. Data message format

The bit-widths shown in Figure 2 represent the field widths used in our proof-of-concept implementation.

The message header provides version information and includes bit-fields which flag additional capabilities and information such as the presence of other data fields, and fused or relayed data. For example, a *stream update request* acknowledgement field is expected to appear in data messages generated by receive-capable sensors.

The composite StreamID field is used to identify the data stream to which a message belongs.

Sequence or timing information is conveyed to allow messages to be correctly ordered and duplicates removed.

The payload field is not interpreted and is opaque to the Garnet infrastructure. This provides a basic level of security and contributes to our security model.

For simplicity, we do not indicate the usual checksums associated with the data messages.

## 5. Critical Design Choices

In this section we elaborate on the rationale behind the major decisions which influenced our design.

### Generality of design

We pursued a general solution which permitted both the receipt of sensor data and the provision of a return path to route control messages. This approach allowed us to consider general issues related to manipulating sensor data streams, which could then be compared to specific, degenerate scenarios, where some subset of the overall functionality was provided.

### Simplicity of sensor requirements

A minimum level of sensor intelligence was assumed to allow for a richer model to be developed, where both simple and sophisticated sensors could coexist.

**Data stream abstraction for managing data** A device-neutral mechanism was required which permitted a low-level of interaction between sensor and consumer in a portable, fine-grained manner.

The focus on *data streams* permits flexible and generic data processing operations to be supported.

### Inferred location data

Access to location data is a refinement which is required to reduce transmission costs when forwarding control messages to sensors. Additionally, such information was required without the active involvement of the sensors. Otherwise, all sensors would be expected to be location aware, which would defy our goal of allowing simple and sophisticated devices to co-exist.

### Generality of location information processing

We avoided placing a location field in the data message header, since it would impose a transmission burden on all sensors, especially those without location awareness. As a compromise, we allow consumer processes to provide location hints instead. Our mechanism offers slightly more generality since a consumer may be able to infer, or otherwise acquire knowledge of, the location of a sensor which is not itself location-aware.

### Delayed delivery decision-making

The StreamID in the data message implicitly identifies the source of the message, while the end destinations are inferred. Deferring delivery decisions allows flexible mechanisms in the fixed network to be used, instead of requiring the sensors to be application aware.

Removing this task from the sensors enhances their simple design and permits an inter-operable, plug-and-play environment to be established where sensors can be deployed into an existing network.

### Simplicity of consumer requirements

With consumer processes accessing data in a mutually unaware manner, attention was focused on defining and developing the middleware services to manage conflicts. This streamlined the operation of consumer processes by removing the need for their direct interaction with, and knowledge of, each other.

This approach allows the use of design patterns [7] to extend the services already offered in a sensible manner, and leads to a simple, reusable framework for processing the data streams.

## 6. Novel Aspects of Architecture

In this section, we emphasise the most significant features of this work. A complete review of our architectural goals with details on all components is available from [17].

### Support for a rich sensor network environment

Applications may route messages back to suitably capable sensors. This allows the consumers to interact with the sensors as well as dynamically control the data streams.

### Consumers are mutually unaware of each other

This simplifies issues related to their direct interaction but requires system-level services to resolve conflicts.

### Heterogeneous sensor capabilities

Few demands are imposed on the operation and behaviour of sensor nodes. This contributes to the reality of continual system evolution which results in the deployment of heterogeneous sensors.

### Ease of extensibility of the original platform

Consumer processes can generate additional data streams and could form an essentially arbitrary graph of consumer processes and data streams over the Garnet middleware. In practise, this graph would be more structured and would be expected to form a hierarchy where lower level consumer processes generate derived streams of data which are consumed by higher-level consumers.

### Approximate overview of sensor configuration

The resource manager acquires an approximate overview of the sensors' configuration. This allows admission control decisions to be made, and is necessary given the potential for conflicting consumer requests.

### Approximate overview of key consumers

The super coordinator acquires an approximate overview of the behaviour of key consumer processes. From this "nearly correct" information, we have identified scope for its involvement in the dynamic control

of the sensors. This behaviour stems from its ability to predictively anticipate changes and invoke the services of the resource manager, reducing the effect of latencies arising from message-handling. Significantly, this provides opportunities for user-defined policies to be enacted, leading to a policy-driven middleware infrastructure.

#### Delayed delivery and distribution decisions

An address-free routing mechanism allows data delivery and distribution decisions to be delayed until the data arrives at the fixed network. This permits application-level and network-level knowledge to be harnessed, to best contribute to overall flexibility.

### 6.1. Discussion

The major benefits offered by Garnet relate to the ease with which simple and sophisticated sensor devices can co-exist. Providing middleware services to manage the sensor data streams also reduces the burden on application developers and simplifies the operation of consumer processes.

A major part of our ongoing work relates to the predictive aspect of the operation of the super coordinator as a means of determining consumer state and reducing latencies in detection and actuation.

We believe that such information would be invaluable in critical environments and we are actively developing suitable models which could be applied to the management of a complex water course. In such a scenario, the ability of the super coordinator to anticipate changes to water bodies and preempt actuation requests is expected to be significant.

### 7. Related Work

In [15], a sensor network for habitat monitoring is made generally accessible for querying data, but differs from our approach for sensor-level and network-level configuration since in [15], only short-range, direct diagnostic level network interfacing is permitted. Unlike Garnet, no open, application-level involvement in the network configuration or the regulatory middleware is provided.

The dynamic variation in consumers and our desire for multiple receivers requires that the sensor nodes do not participate in the routing of the data. Our approach differs from the data-diffusion technique in [13], which permits nodes to judge the best hop for data routing. Garnet transparently supports such node level activity, although no means are currently provided to process and route such multi hop data to its source. With node-level agreement on data forwarding and relaying, such activity would be transparent to Garnet.

Elson and Estrin [5] provide quantitative experimental measurements for reducing energy costs when using a Random Ephemeral Transaction Identifier (RETRI) for data communication in sensor networks. Their RETRI technique

reduces the cost of data transmission by using fewer bits to identify a transaction, instead of the larger pre-defined sensor and stream identifier header fields used in our message format. Their approach scales with the increasing transaction density and not the sheer size of the network. At a first glance, their assumptions of transaction locality suggest possible adaptation of their RETRI technique as a means of reducing the energy cost of data transmission in Garnet environments. However, because Garnet depends on unique consistent stream IDs, the ephemeral nature of the RETRI identifier renders their technique inappropriate. However, the stream update request identifier issued to consumer processes and used in sensor-level acknowledgements in Garnet messages is loosely comparable to a RETRI.

Steere et al [18] describe CORIE, a real-time Environmental Observation and Forecasting System used to monitor a large river estuary and plume, whose sensors are used to drive 2-D and 3-D models. Unusually, CORIE's sensor nodes are capable of generating megabytes of data per second, well above the range expected for "typical" sensor networks. Despite the up-scale performance characteristics of CORIE, the authors assume that at most a few competing applications will run concurrently. This suggests a close coupling between the output data and the applications, a shortcoming that Garnet is designed to address.

Madden and Franklin [14] developed the concept of Fjords to resolve the data-management issues associated with querying continuous sensor-streams and static data. They advocate the use of sensor proxies to permit a set of queries to operate over the same sensor stream, and show that the sharing resulted in significant improvements to their ability to handle simultaneous queries. Both the Fjord and Garnet architectures share the notion of separating the consumer of the data from its source. Fjord sensor proxies also share similarities with the use of our resource manager in adjusting sensor output based on user demand.

### 8. Future Work

We are now extending the deployment of the architecture to an environment where the sensor hardware is simulated by a mixture of iPAQs and notebook PCs with the IEEE 802.11b [11] network used as the wireless medium.

We aim to apply these techniques in the context of a large scale project such as EQUATOR [19], which relies on the instrumentation of outdoor environments to fuse physical and digital environments. The IDEAS project [12], with system-on-a-chip control of lab-on-a-chip monitoring sensors, also provides a suitable testbed for the Garnet prototypes, allowing interaction with intelligent sensors in realistic but controlled environments.

Building on our work to-date, we are now exploring several interesting problems and key generalisations of our initial model which include:

- The development of predictive mechanisms to fore-

see the resource needs of the system. This would contribute to the effective planning of resource utilisation. The use of both application-supplied state transition information and system-inferred changes to data usage patterns are being explored.

- Codification of sensor constraints via the development of an expressive language. This would facilitate the operation of the resource manager in automatically enforcing such limits.
- Exploration of the implications of supporting multi-hop routing within the sensor network. Single-hop routing assumptions allow a number of simplifications to be made which need to be revised when considering multi-hop routing. Such issues arise if the source of relayed data is not immediately accessible or available when transmitting control messages.

Initial support has been provided by tagging the message header to reflect multi-hop and relayed data messages to facilitate intelligent processing decisions.

## 9. Conclusion

We have explored the problem of providing middleware support for application programmers who wish to utilise data streams which originate in wireless sensor network environments. The motivation for this work is derived from the increasing affordability, availability and scale of future sensor networks and the need to simplify the development effort required by disparate software engineering teams, while maintaining separation of concerns of network operators, hardware designers and programmers.

By approaching the problem from the architectural level, common, low level functions were factored out and provided as middleware services. The resulting architecture of Garnet is sufficiently general to support the properties identified as necessary to adequately fulfil the middleware role.

By offering our collection of system services such as the Receivers, Filtering and Dispatching Services, Resource Manager and Orphanage, a number of useful properties and novel features were made available.

Garnet has been initially implemented using Java. We are now adding wireless devices to more faithfully model mobile sensors and allow us to further explore practical issues of scaling and heterogeneity.

Novel features of Garnet include: the abstraction of location, with inference of location from reception data combined with application level hints; embedding of sensor management policy to allow mutually unaware applications to tune the behaviour of unwittingly shared sensors; support for trusted applications to provide advance warning of changing needs and override sensor management policies; and a high-level abstraction of data streams supporting end-to-end encryption.

## Acknowledgements

We wish to thank our colleagues in the Networked Systems Measurement and Control Group and the anonymous reviewers for their helpful comments and suggestions.

## References

- [1] M. Bhardwaj, S.-H. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. Upper Bounds on the Lifetime of Sensor Networks. In *Proc. ICC 2001*.
- [2] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann. Scalable Coordination for wireless sensor networks: Self-Configuring Localization System. In *Proc. ISCTA 2001*.
- [3] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In *Proc. SIGCOMM 2001*.
- [4] A. Cerpa and D. Estrin. Ascent: Adaptive Self-Configuring sEnsor Network Topologies. Technical Report UCLA/CSD-TR 01-0009, May 2001.
- [5] J. Elson and D. Estrin. Random, Ephemeral Transaction Identifiers in Dynamic Sensor Networks. In *Proc. ICDCS-21*.
- [6] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *Proc. ICASSP 2001*.
- [7] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, 1994.
- [8] L. Girod and D. Estrin. Robust range estimation using acoustic and multimodal sensing. In *Proc. IROS 2001*.
- [9] W. R. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proc. HICSS '00*.
- [10] W. R. Heinzelman, A. Sinha, A. Wang, and A. P. Chandrakasan. Energy-Scalable Algorithms and Protocols for Wireless Microsensor Network. In *Proc. ICASSP 2000*.
- [11] IEEE Std 8802-11:1999(E). *IEEE Standard for a Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. August 1999.
- [12] Institute for System Level Integration (ISLI). Integrated Diagnostics for Environmental and Analytical Systems (IDEAS) Project, November 2001. <http://www.sli-institute.ac.uk/research/ideas.htm>.
- [13] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. MobiCom 2000*.
- [14] S. Madden and M. J. Franklin. Fjording the Stream: An Architecture for Queries Over Streaming Sensor Data. In *Proc. ICDE 2002*.
- [15] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *Proc. WSNA 2002*.
- [16] R. Min, M. Bhardwaj, S.-H. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. An Architecture for a Power-Aware Distributed Microsensor Node. In *Proc. SiPS '00*.
- [17] L. St. Ville. Garnet Architecture, November 2002. <http://www.dcs.gla.ac.uk/~lsv/garnet.html>.
- [18] D. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole. Research Challenges in Environmental Observation and Forecasting Systems. In *Proc. MobiCom 2000*.
- [19] University of Nottingham. EQUATOR Interdisciplinary Research Challenge, November 2001. <http://www.equator.ac.uk>.