Node and Network Architecture for Wireless Pico Systems

Yiming Wang[1], K. Sivansen[2], and Tughrul Arslan[1,2]

[1]*University of Edinburgh, School of Engineering & Electronics, Edinburgh EH9 3JL, Scotland, United Kingdom.*

[2]*Institute for System Level Integration, The ALBA Campus, Livingston, EH64 7BH, Scotland, United Kingdom.*

Abstract—PicoWireless Networks [1] are being proposed in applications such as environmental monitoring, intruder detection and smart buildings. However, the two main challenges of PicoWireless Networks are energy and bandwidth. This paper describes the design of cluster formation and optimization of network management in order to prolong the life of Pico Wireless Sensor Nodes and increase their bandwidth. The paper also describes the node architecture targeting an SoC platform.

I. Introduction

Recent advances in MEMS technology, RF circuitry and signal processing have brought this idea of small, low-power wireless devices ever closer to reality [1]. Pico Wireless Sensor (PWS) nodes [1] are very small devices, typically a few centimetres^3 in volume. Typically a network with hundreds of these devices will be used in each application. Potential uses of such devices are unlimited; environmental monitoring, intruder detection and smart buildings are to name a few [3].

The two main challenges in designing PWS nodes are energy and bandwidth. The available energy within such nodes is limited. Therefore every aspect of the design process must also be considered form the energy prospective to extend the device lifetime. The bandwidth for wireless communication is also limited. Therefore there is a need for innovative solutions to effectively utilise the available bandwidth [5].

Currently significant amount of research effort has been directed towards this area. The PicoRadio [1] and µAmps [2] projects, for instance, aim to integrate sensing, processing and radio communication onto a microsensor node. Other research work aims to integrate both SoC and laboratory on chip technologies in wireless devices used in "pico-like" applications [6].

Research in this paper aims to add another dimension 'motion' to wireless nodes. A cluster-based approach is chosen as the backbone network architecture. In this approach nodes send their data to a central cluster-head node, which forwards the data to the base-station. The cluster-head acts as a local control centre saves energy by time-sharing between all nodes. The network is robust since the cluster-head role is not fixed. Intra-cluster data are sent directly from one node to the next via the cluster head, whereas inter-cluster data and cluster to base station data are routed through the cluster-head nodes.

The rest of the paper is organized as follows. We introduce the network design in Section II. Implementation and results are presented in Section III. In Section IV, we conclude the paper.

II. Design of PicoWireless Network

A. Cluster Formation

This research aims to add another dimension 'motion' to a wireless node in a Pico Wireless network. This will expand the application domain of such networks to others such as habitat monitoring and plant monitoring.

In the network proposed in this paper, all the nodes are identical except for the node identification number (NIN), which is unique to each node. Nodes can become ordinary nodes or cluster-head nodes. The cluster-head acts as a local control centre, and to save energy the cluster-head's role is time-shared between all nodes. This enables robust networking with point-to-point connectivity. Intra-cluster data are sent directly from one node to the next via the cluster head, whereas inter-cluster data and cluster to base station data are routed through the cluster-head nodes. This design allows for increased system capacity by reducing interference.

If the network is to be of any use, it must form efficient clusters. At the beginning (of the system life) all nodes attempt to join the best cluster; best cluster is the one in which participation requires the least amount of energy. On joining a cluster, the nodes obtain the operating parameters from the cluster-head and schedule their tasks accordingly. To avoid the initial overload of channels, nodes become active at pseudo-random intervals.

If a suitable cluster is not available, nodes will attempt to form clusters by becoming cluster heads. To elect itself for the cluster-head position, a node must satisfy a set of constraints. One of the main duties of the cluster-head is to forward data to base-station; therefore it must "see" the base station. Also a good cluster-head should be able to see the majority of nodes in the cluster. Any contention between nodes to become cluster-head is resolved randomly. The cluster-head position will expire after a certain time interval1.

After establishing itself as cluster-head, the node (cluster-head) requests operating parameters form the base station. Other nodes can request membership and may be allowed to join when nodes present in the cluster is less than the maximum amount of nodes allowed per cluster. The cluster-head will broadcasts operating parameters on request to member nodes.

This process of cluster formation is repeated if nodes find themselves not in cluster due to movement. Since nodes are mobile and SDMA is used, nodes are given unique identification to avoid confusion.

Employing power control forces optimised clusters and enables further energy savings. The output power of the transmitter is divided into eight levels, where level 1 is the minimum and level 8 is the maximum. When a node wants to join a cluster, it starts by sending the request to join at level 1. If there is no response at all the frequency bands, the node then increases the power level by 1 step. This is repeated until a cluster is found. This method ensures that node within a cluster are radio near, hence consuming minimal energy for radio transmissions.

B. Node Architecture

The chosen architecture of the PWS node [5] consists of a microsensor array, general-purpose microprocessor, memory, the PWS IP, radio DSP and radio hardware. The microsensor interface converts the desired environmental properties into digital values. The operating system (OS) executes application specific algorithms on the microprocessor to process the data. As well as executing application algorithms, the OS is responsible for scheduling the power-down and active periods of the microsensor array and the microprocessor. The OS is stored in the ROM memory and moved to RAM when required.

The application algorithms implement a variety of user tasks such as data processing and data aggregation. These algorithms are also stored in the ROM memory and moved to RAM during execution.

This architecture is a compromise between the low power requirement and flexibility. The functions that are independent of the application, such as network management are realised in hardware while application specific components such as the data processing algorithm are implemented in software. This allows the PWS node to be customised for various applications simply by changing the software.

III. Implementation and Results

The mobile PWS system was implemented with an SoC architecture which includes a microprocessor core to form the complete node. In addition, the system consists of the following modules: control, memory, application-link layer's interface, join algorithm and operation parameter transfer algorithm. These modules were successfully synthesised using the UMC .18um technology. All modules were designed with signal and clock gating in order to minimize power where possible.

A. Testing the Join-Cluster algorithm

The algorithm starts by sending off requests to join a cluster. The requests may not get a reply, get the correct reply or get an incorrect reply. Depending on the type of reply, the algorithm then returns it response.

In order to fully test the functionality, it is essential to cover all frequency bands and power levels which will consume a large amount of time. An alternative is to use random testing and to test the join algorithm. The tests simulated following three scenarios: no reply arrives in time, the correct reply arrives after a random amount of time, or an incorrect reply arrives after a random amount of time.

Starting from power level zero, the frequency bands should be increased from start-band + 1 to end-band. When the end-band is reached, power level is increased by one and the frequency band is reset to start-band. The above steps are repeated until power level eight is reached and the response is set to '01' to indicate no clusters are available.

B. Lifetime of clusters

The cost of forming clusters in terms of energy is high. Therefore once formed, clusters should last as long as possible. On the other hand if a single node continues to be a cluster-head, it will die-out quickly, since being a cluster-head is more energy demanding. Therefore it is necessary to derive the optimal cluster head period. To keep the models simple, the following section ignores node mobility.

If a node is allowed to be a cluster heads continuously, it will die-out after certain amount of time tnm (the lifetime of cluster-head node) and another node will takeover as cluster-head.

In our cluster formation, the cluster-head is not fixed, the quality of the network is at maximum (1) from the starting point to T_L (T_L is the lifetime of nodes) and falls to zero after this point. This is diagrammatically represented in figure 1.

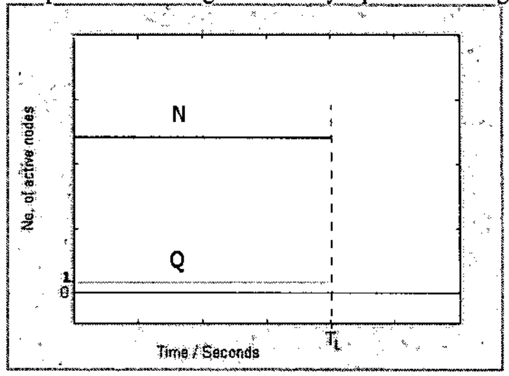

Figure1. Quality in a fair network

IV. Conclusions

In this paper, we have described the design of a specific PicoNode Wireless Network and associated node architecture using SoC technologies. We concentrated on Cluster Formation. The primary advantage of the PicoWireless Network is mobility. However, mobility has its own constraints while freeing from wires. Especially available bandwidth for wireless communication and lifetime of nodes are limited. From our simulation results, the specific network design improve the efficiency of bandwidth and power usage, it also prolongs the lifetime of nodes.

References

[1] Jan M. Rabaey, M.Josie Ammer, Julio L.da Silva Jr., Danny Patel, Shad Roundy, "PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking" Article featured in the July, 2000 Issue of Computer Magazine.

[2] Anantha Chandrakasan, Rex Min, Manish Bhardwaj, Seong-Hwan Cho, and Alice Wang, "Power Aware Wireless Microsensor Systems", Keynote Paper ESSCIRC, Florence, Italy, September 2002

[3] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, T. Tuan, , "PicoRadios for Wireless Sensor Networks: The Next Challenge in Ultra-Low-Power Design", Proceedings of the International Solid-State Circuits Conference, San Francisco, CA, February 3-7, 2002

[4] K. Scott and N. Bambos. The Self-Organizing Wireless Adaptive Network (SWAN) Protocol for Communication Among Mobile Users. In Proceedings of the IEEE

[5] Kajanandth Sivansen, Application Specific Pico Wireless Networks, MSc Thesis in Institute for System Level Integration.

[6] Aydin, N.; Arslan, T.; Cumming, D.R.S.; "Design and implementation of a spread spectrum based communication system for an ingestible capsule", 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002.