Universität Karlsruhe (TH)
Institut für Telematik

TELEMATICS TECHNICAL REPORTS

A Framework for Network Mode Control in Wireless Sensor Networks

Christian Prehofer, Bernhard Hurler, Qing Wei, Martina Zitterbart
prehofer@docomolab-euro.com,hurler@tm.uka.de,
wei@docomolab-euro.com,zit@tm.uka.de

December, 21th 2005

TM-2005-5

ISSN 1613-849X

http://doc.tm.uka.de/tr/

Institute of Telematics, University of Karlsruhe
Zirkel 2, D-76128 Karlsruhe, Germany

This page is intentionally left empty.

Mode Control in Wireless Sensor Networks

Abstract. In many areas of pervasive networking, e.g. assisted living, intelligent house, or smart office environments, wireless sensor networks (WSN) can be used to efficiently monitor, control, or serve as alarm systems. Depending on the context, the network may be used for data collection by different applications with widely different requirements, e.g. energy efficient monitoring or highly reactive alarm systems. This paper introduces the new concept of network modes, which describe different operational modes of a WSN to serve different applications in an optimal way. We argue that there are a few basic modes, i.e. energy mode, fast delivery and redundant mode, which can be used for various applications. In this way, the applications only specify their needs, but do not need to care about the detailed network operation. We present a generic mode control layer, which can be used with different mode implementations and also performs adaptive mode switching depending on requirements of the target mode, e.g. energy efficient switching or fast switching. We also show how different distributed algorithm for mode implementation and mode switching can be used in the framework. In a case study we prove the feasibility and demonstrate the functionality of our framework.

1 Introduction

In many areas of pervasive networking, e.g. assisted living, intelligent house, or smart office environments, wireless sensor networks (WSN) can be used to efficiently monitor, control, or serve as alarm systems. Depending on the context, the network may be used by different applications for data collection with widely different requirements, e.g. energy efficient monitoring or highly reactive alarm systems. While most of the research on sensor networks aims to optimize networks towards specific criteria, such as energy-efficiency, we present a new concept for dynamically adapting sensor networks to different and possibly conflicting application needs. This paper introduces the new concept of network modes, which describe different operational modes of a WSN to serve different applications in an optimal way.

The motivation for this is as follows. WSN applications differ from conventional applications and equally also data communication in WSNs differs. First, WSN applications are typically distributed over a large number of sensor nodes and their communication needs are often known prior to their time of execution. For instance, networks for collecting data have a communication pattern which is often clearly defined in advance. Second, communication in WSNs is usually characterized by so-called data-centric communication. This means amongst others that data from multiple sources and sinks, are aggregated, transformed, or otherwise processed within multi-hop connections. Thus, there are various algorithmic possibilities to implement data transfer in WSNs whereas the actual method is highly dependant on the optimization goal. Both a-priori known applications and their distributed and complex communication behavior allow and demand for a different approach to offer selectable and appropriate communication characteristics in WSNs.

This is often the case for e.g. the large field of environmental monitoring (observing animals, spotting forest fires, etc.) These kinds of WSNs are characterized by a high number of uniform sensor nodes executing the same task. Thus, also the network behavior remains the same and is optimized for the specific purpose (energy efficiency, high responsiveness, etc.) WSNs with heterogeneous applications can be found in the area of assisted living, intelligent house, or smart office environments for example.

A main motivation of this paper is that a small number of modes can be sufficient for a large variety of applications. From this observation, we propose a new network mode control layer which adapts the network to different modes, according to the active applications. It enables WSN applications to control and change intra-application communication characteristics of entire affected parts of the network. We assume for this that each network has a dedicated number of applications which can announce their communication needs in advance. Based on this, we devise a generic mode layer which implements modes in an effective way. We think that this provides another step towards middleware concepts for pervasive sensor networks [10].

The operation of this new network mode control layer is not limited to one network link or route, but often affects whole parts of a network engaged in one application. This layer relieves the applications from the complexity of establishing, maintaining, and switching communication characteristics. This includes for instance dealing with conflicts between several applications, breaking links, or changing communication characteristics. This is now accomplished by a separated layer by which applications can control "overall" communication characteristics of all participating nodes and connections.

Introducing network modes as a new concept in WSNs leads to a number of challenges to be addressed. For instance, we generally aim to reduce the overhead of switching between modes, as switching may occur frequently in the network. A main insight is that the switching has to fulfill different requirements, depending on the target mode. For instance, switching to energy efficient mode can be done slowly and energy efficient, while other modes require fast switching. In summary, the main issues addressed in the paper are the following:

1. Can we identify modes which are generally useful for many applications?
2. How can the mode concept be realized in a WSN in a flexible way?
3. How can we switch the WSN efficiently between different modes?
4. Is the mode concept feasible in real WSNs?

In the following, we introduce three general-purpose network modes. Section 2 introduces a framework for network modes enabling its application in WSNs. Switching network modes is a very important issue in this framework due to its high impact on the operation of the modes itself. Its interplay with network modes, especially its effects on the mode to be switched to, is discussed in section 2 and the integration into the framework is described there. To analyze feasibility and usability of network modes we implemented an instance of the framework in a real WSN described in section 3. Section 4 gives an overview over related work addressing the issue of quality of service and network characteristics in WSNs. In section 5 we conclude the paper with a short summary and the main achievements of our work.

1.1 Application scenario and assumptions

This section details our scenario and summarizes the main assumptions. We assume applications specify a desired, periodic data flow pattern with specific properties. This includes the data sources, sinks and the data to be transmitted including periodicity. The specification may be generic in the sense that the specific sources may be specified, e.g. all temperature sensor nodes in a room, but need not be named explicitly. Furthermore, possible data aggregation may be specified to reduce the number of messages to be sent. Hence the applications do not need to know the actual paths in the network and possible aggregation nodes. We consider this data collection as the main task of the sensor application. The applications can in addition request specific optimizations, as we express by modes below.

We assume for simplicity that only one application is active at a given time and operates on the complete network. These limitations and potential future generalizations will be discussed later.

1.2 Network modes

In the following, we introduce our concept of modes, which implement the above application specification in different ways. Network behavior of a mode is determined by several factors. Individual nodes of a WSN have different capabilities, like processor sleeping cycles, radio device activity, energy status, or current connections. Communication links between individual nodes also vary in characteristics like link quality, bandwidth, or (perhaps energy dependant) communication range. Similar to single links entire routing paths have variable characteristics including more sophisticated mechanisms like in-network data processing, aggregation, or cluster-based routing. All these characteristics for individual *nodes*, *links*, and *routing* protocols can be constant or dependant on current situations. They will be referred to as *network parameters* in the remainder of the paper. The interaction of several network parameters results in a certain network behavior. A set of network parameters and their resulting network behavior is summarized to one *network mode*. Thus, a network mode is a defined network behavior achieved by setting up specific network parameters accordingly.

In the following, we explain the requirements of the modes and show typical methods to implement them. We do not assume fixed or generic implementations, but rather build a framework (see Section 2) which can accommodate several implementation options.

1.2.1 Energy mode

The energy mode optimizes energy consumption in order to aim for a long-lived network. This is achieved by several schemes for saving energy. Single nodes may choose appropriate sleeping cycles and the network uses energy-efficient routing schemes. Regarding the network parameters the energy mode is characterized as follows:

1. allow energy-saving mode (e.g. sleeping) of nodes tolerating reduced responsiveness and speed,
2. reduce communication (i.e. by data aggregation, avoiding retransmissions, lower frequency of data collection),
3. allow modification of network topology.

1.2.2 Fast mode

The fast mode relaxes energy constraints. Within energy mode the WSN is supposed to save as much energy as possible, whereas in fast mode the main purpose is to provide communication with low latency. The fast mode is characterized as follows:

1. prevent use of power-saving and sleep mode of nodes,
2. no data aggregation,
3. direct connection with least intermediate nodes (long wireless links allowed),
4. data forwarding with minimum delay.

1.2.3 Reliable mode

In reliable mode energy constraints are relaxed even more. In addition to the no aggregation parameter redundant data forwarding is introduced. Thus, data transfer is high-reliable and fast. The following network parameters are set in reliable mode:

1. prevent use of power-saving and sleep mode of nodes,
2. no data aggregation,
3. allow redundant data,
4. use redundant routing paths.

1.3 Example scenario

In the following we introduce a small example scenario from the area of assisted living to demonstrate the usage of different network modes.

Figure 1 shows an overview of an exemplary scenario. It is composed of three distributed WSN applications: body monitoring, health alert, and fire alarm.

The basic task of the body monitoring is to monitor a person's health condition. Several sensor nodes are responsible for sensing therapeutic data, i.e. blood pressure, heart beat, and motion. All sensors measure the respective sensor value once a minute. The compound values of all three sensor nodes will be stored in one of the nodes of a distributed data storage, which consists out of several distributed storage nodes. To minimize communication the health values are aggregated in one of the sensor nodes and sent to one of the storage node in a round robin scheme.

If the above described body sensor nodes detect a dangerous health condition (e.g. heart beat is too high), the heath alert application is executed. In contrast to normal function every sensor node sends its values more often (once in a second), and data are not aggregated any more but sent directly to the distributed storage. Additionally, an internal acoustic alarm and an external emergency call system are informed.

The third application observes the room temperature with several (in this case three) temperature sensors. As soon as the value of one sensor indicates a potential fire the respective sensor initiates the fire alarm application. In this case every temperature node chooses a storage node to send and store its temperature values. All storage nodes receiving such data send an alert message to the alarm nodes.

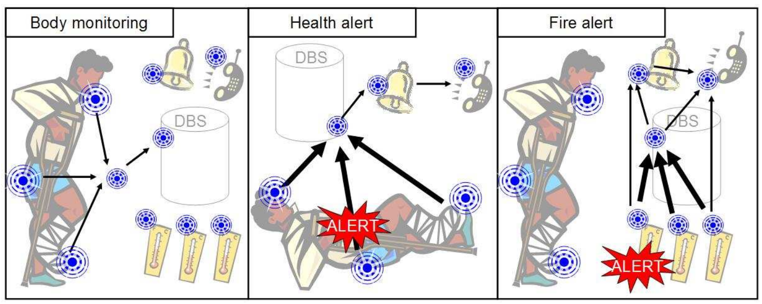

Fig. 1. – Exemplary scenario consisting of three distributed WSN applications

In this small example scenario applications run distributed on some or all nodes of the WSN. Dependant on the current situation and application the WSN has to offer different network behavior. The body monitoring application is a long-term task and requires energy efficient communication (with no necessity for fast or reliable connections) and so would use the *energy mode*. The health alert and fire alarm application on the other hand require fast and/or reliable data connection and, thus, would use *fast mode* or *reliable mode* accordingly. Using fast or reliable mode in this scenario implicates an appropriate switch to these modes, too. E.g. switching from energy mode to fast mode due to an alert situation requires a *fast* switch to the *fast* mode. This exemplary scenario shows the necessity of different network behavior (accomplished via different network modes). Also, switching modes should be done in different ways depending on the target mode.

The following section introduces a framework addressing the use of network modes and the aspects of mode switching.

2 A Framework for Network Mode Control

In this section, we introduce our architecture and generic framework for controlling modes and switching between them in an efficient way. The framework is kept generic in the sense that different algorithms implementing modes, e.g. clustering algorithms, and specific mode switching techniques can be selected. We assume that the actual mode control is typically custom made by plugging in algorithms, which we have discussed above. Another alternative is pre-configuration of the parameters to implement a specific mode, which is often a suitable option, e.g. static networks.

This means that we do not aim for a universal software layer which consists of many configurable algorithms and which can handle many different configurations. As we target embedded, limited devices, we only plug in specific code, in order to keep the software size and complexity low for each deployment. This is more in the line of product line software [1] and not of monolithic middleware approaches.

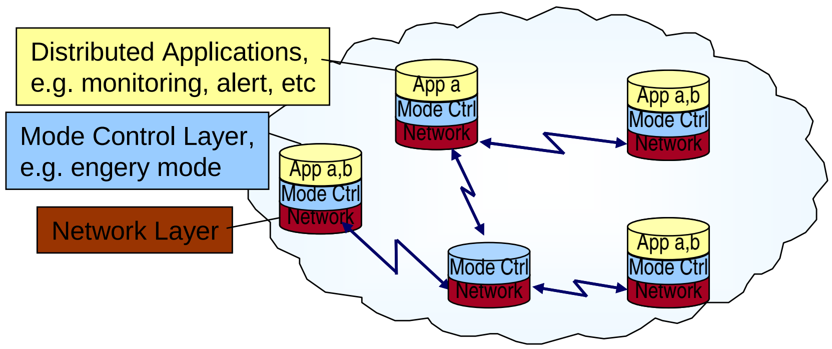

Fig. 2. Network and Application Architecture

While the actual performance of a mode depends on the chosen algorithms, the general objectives of our mode control layer are the following:

– Flexibility with respect to application requirements.
– Performance of mode switching in terms of resources used for mode switching, including time, energy, signaling overhead, etc.
– Consistency and convergence of the mode switching, i.e. all nodes should eventually have the same mode. As we work in a distributed environment, temporary inconsistencies are possible.

It is important to notice that the latter two requirements are actually dependent on the target mode to be switched to. For instance, when switching to a mode used by an emergency application, energy usage and consistency within the network are less important than fast reaction to the mode change trigger.

Our network architecture is shown in Figure 2. We assume a wireless network with several nodes and several applications, which run in a distributed way on these nodes. For instance, a monitoring application may reside on several nodes, where some nodes measure certain sensor data and a monitoring node collects these data. While several applications may be active at the same time, we assume that only one mode is active and hence the network is optimized for only one application.

Below the application layer we introduce the mode control layer, which controls the network modes. This also includes mode switching and conflict handling, e.g. if different nodes issue inconsistent requests. This is done by priorities between modes.

This control layer is present on all nodes which are involved in mode switching. Its task is to map the applications' requests for network modes to appropriate configurations of the networks. The application will specify the key parameters of the modes, e.g. data sources and sinks, but do not need to know the details of the network, e.g. network topology.

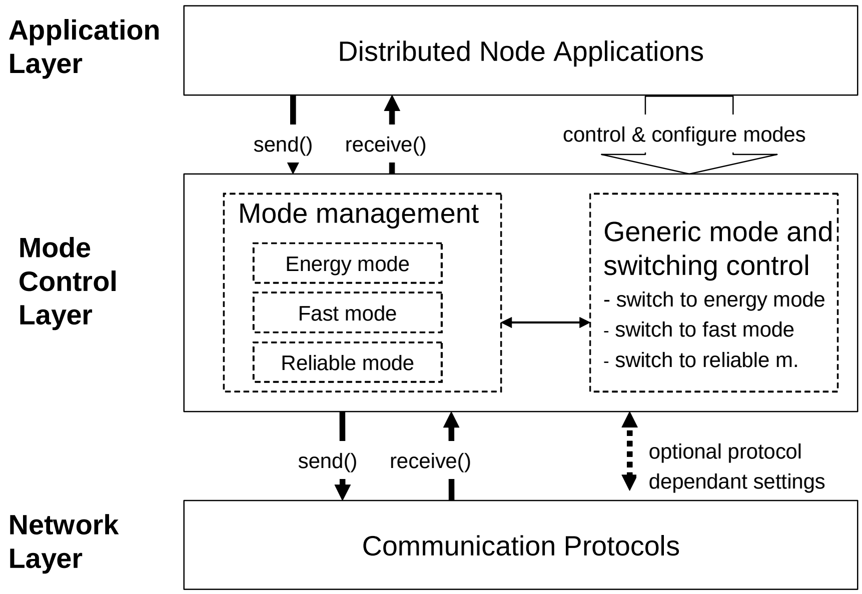

Fig. 3. Node Architecture with Mode Control Layer

The mode control is implemented in a distributed way, without any central control. Furthermore, the interface to the applications is also not fixed to a specific node. Applications can trigger mode changes from different nodes. For instance, a small temperature sensor can trigger a mode change.

The initial configuration and deployment of the mode layer is done beforehand and must be done consistently at all nodes. This will typically go along with application deployment. This means that appropriate implementations of the modes are selected and parameterized. Each application has to select the mode it prefers and has to set parameters appropriately.

In summary, our approach consists of the following three phases, which depend strongly on the chosen mode implementations.

1. In the deployment phase of the network, the appropriate mode and switching implementations have to be selected, put together and deployed. In addition, appropriate configurations of the modes suitable for the network are installed, unless these parameters can be determined by dynamic mode algorithms. For instance, with advance configuration of nodes, the aggregation nodes have to be determined and sleep cycles be agreed. Furthermore, the desirable switching mechanisms have to be selected for each possible switching. As shown in Table 1, we have six cases and appropriate switching methods have to be defined.
2. In the second, startup phase, the applications can configure the mode layer with parameters which are not set at the deployment phase. For instance, a flexible energy mode implementation can be configured with different data sources and data sinks as well as a data flow model to collect the data.

3. In the operational phase, the applications can trigger mode changes and the mode layer has to implement these requests.

Figure 3 gives a detailed view of the mode control layer in our node architecture. At each node, the mode control consists of the mode management and the mode and switching control modules, which are independent tasks.

The mode management consists of the specific algorithms to implement the modes. The algorithms depend on the desired mode implementation and must evidently be consistent at all nodes. In a basic implementation, the role is configured in advance and activating the mode will implement this role. For instance, aggregation nodes and sleep mechanisms are defined to implement the energy mode. In more flexible implementations, the mode implementations will dynamically compute the role. For instance, aggregation nodes can be determined by clustering protocols which determine cluster heads used for data aggregation. In this dynamic version the cluster algorithm may run independently of the mode management. Depending on the desired setting, the cluster heads may be computed proactively or on demand.

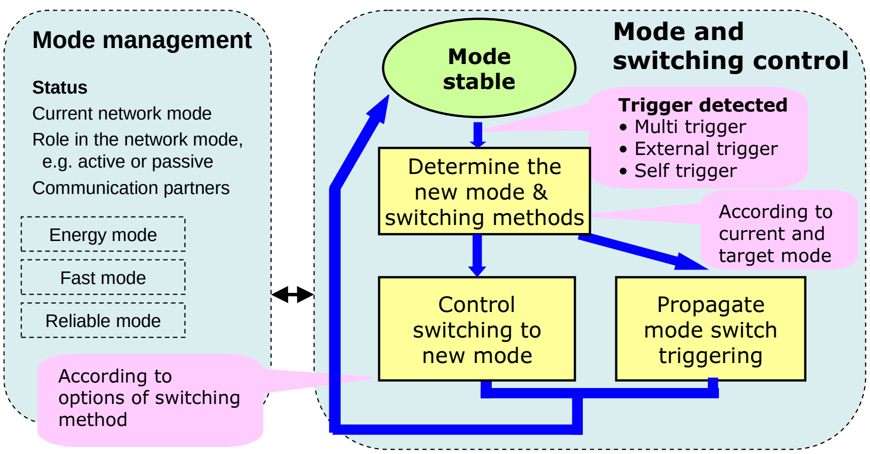

Fig. 4. Mode Management and Switching Modules

Figure 4 shows the mode management and switching modules. The mode switching and control module is in charge of controlling the mode status and of the switching process. We illustrate the generic procedure by the flow chart shown in Figure 4. In the normal state, the mode is stable and waits for requests triggering mode changes. These are typically external events, but can also be internal timers if mode switching is controlled by time-outs. In case of a trigger, which includes multiple trigger events, the next step has to determine the target mode and the switching methods. This first of all means to solve conflicting requests from different nodes by priorities. Furthermore, some switching may be lazy and not the first trigger event will cause a switch.

The mode switching consists of two independent procedures. The propagation module shown on the right has to propagate the switch event to other nodes. The switching control module performs the actually switching within the node by triggering the mode algorithms appropriately. The actions performed in both of these modules strongly depend on the target module, and of course on the existing module.

The module for propagation of mode switch triggering has a separate execution thread, because triggers may be sent repeatedly and in some cases triggering has to be delayed if other nodes are in sleep mode. The mode triggering also depends on the target mode:

- When switching to the fast mode, triggers should be propagated fast.
- When switching to energy efficient mode, the switching itself is not time critical and extra wake-up triggers for other nodes should be avoided.
- When switching to the redundant mode, consistency is most important and e.g. messages may be sent repeatedly using acknowledgements.

The actual switching takes place in the control switching module. In simplified terms, it has to trigger the appropriate mode implementation and de-activate the current one. As discussed above, we also optimize the switching to the requirements of the target mode. Mode switching means to re-organize the data distribution path in the network, which allows several target dependent optimizations that are presented in Table 1. For instance, when switching from fast to reliable mode, the current paths may be kept although they may not be optimal in terms of path independence.

Table 1. Switching optimizations depending on target mode

Switch from Switch to	**Energy mode**	**Fast mode**	**Reliable mode**
Energy mode		– Reorganize network topology	– Remove redundant communication paths – Reorganize network topology (if beneficial)
Fast mode	– Ignore intermediate nodes within radio range		– Remove redundant communication paths – Ignore intermediate nodes within radio range
Reliable mode	– Keep original aggregation – Add redundant path	– Keep original paths – Add redundant paths	

3 Case study

To analyze the usability and feasibility of the network mode framework, we have implemented a specific instance of the framework in a real WSN with a small number of sensor nodes. The scenario described in Section 1.3 contains several distributed applications each with different requirements for the network behavior. It includes three different WSN applications, each of them distributed over several sensor nodes, and each application requires different network behavior including the need for energy efficiency, fast data transfer, and reliable communication. Figure 5 shows an abstract overview of the participating sensor nodes and the data flow of each application. It is explained in more detail in section 3.1.

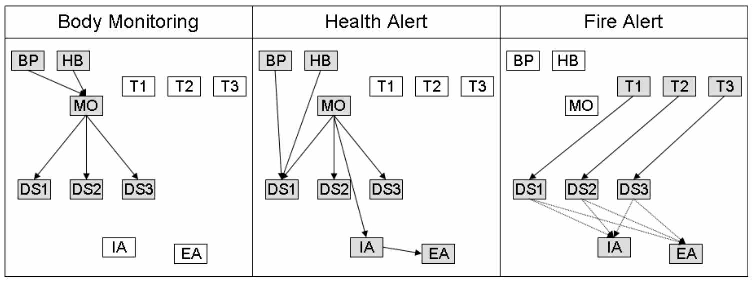

Fig. 5. Participating sensor nodes and data flow of case study implementation

All three WSN applications (body monitoring, health alert, fire alarm) have been implemented in TinyOS. The sensor nodes used in the implementation are standard MicaZ motes from Crossbow Technology Inc. They run with a micro controller ATMEGA128, a radio module based on the IEEE 802.15.4 ZigBee standard, and numerous physical sensors (acceleration, light, noise) and actuators (LED, loudspeaker). In addition to the 11 sensor nodes participating in this scenario a monitoring station was included to enable simultaneous monitoring and visualization of the WSN. It is capturing all sent packets and visualizes all current radio messages and the resulting network topology (see Figure 6 as an example).

The following sections describe how single layers of the framework (distributed application, mode management, and switching control) were implemented.

3.1 Distributed applications and mode management

The scenario consists out of three application and each application runs in one dedicated network mode. Table 2 shows the mappings between application and network mode.

Table 2. Mapping between application and network mode in case study

Application	Network mode
Body monitoring	Energy mode
Health alert	Fast mode
Fire alarm	Reliable mode

Energy Mode

In energy mode (body monitoring application) the network is requested to be energy efficient. Thus, the basic task of sensing bodily data (blood pressure BP, heart beat HB, motion MO) and storing them into a distributed data storage (DS1 to DS3) is done with an aggregation algorithm. We implemented two different algorithms to implement this network mode.

The first implementation has a fixed assignment of the aggregation roles: BP-sensor and HB-sensor are aggregation slaves; MO-sensor is aggregation master. Due to the a-priori role fixing, each node starts its partial task as soon as the mode is in progress. The slaves send periodically once in a minute their data to the aggregation master, which sends the aggregated data (including its own) to the distributed storage nodes in a round-robin scheme.

The second implementation does not assign the aggregation roles in advance but starts with an election phase. The three body sensors elect a master with a priority-based election algorithm. According to certain preconditions (e.g. energy status) the nodes determine their own priority. The node with the highest priority takes on the master role. This election can be repeated whenever priority changes occur.

Both implementations of the energy mode can be interchanged smoothly by replacing the energy module in the framework.

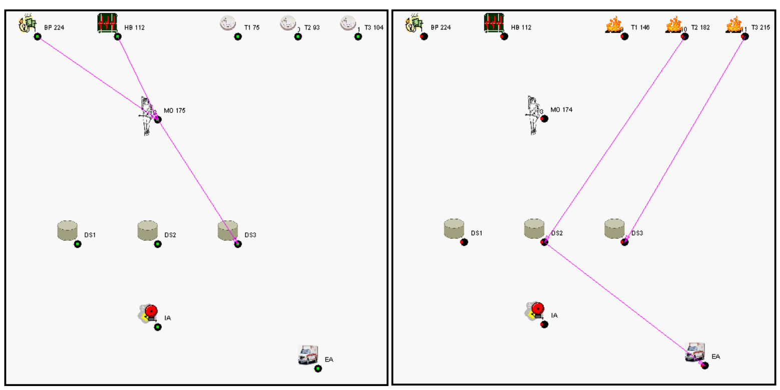

Fig. 6. Snapshots from the case study with energy mode (right) and reliable mode (left)

Fast Mode

In fast mode (health alert application) the network is requested to deliver data as fast as possible. If one of the body sensors indicates an anomalous condition no aggregation schemes is used any more but body sensor data will be sent directly to the distributed storage. Additionally an alarm bell (internal alert node IA) and an emergency call (external alarm EA) are triggered.

Reliable Mode

In reliable mode (fire alarm application) the network is requested to provide a reliable data transport. We implemented this requirement with the help of redundant data transfer. Thus, during a fire alarm indicated by one of the temperature sensors not only one node sends an alert message but all nodes of the distributed storage try to notify the alarm nodes.

3.2 Switching control

The demo implementation runs autonomously in all three modes and switches among them according to physical frame conditions. Section 2 discussed thoroughly how switching to another mode affects the performance of the target mode. Therefore we set a high value on appropriate switching procedures in the switching control module.

Switching control for energy mode
Monitoring of bodily data is seen as the standard application in this WSN running in energy mode. Thus, switching to this application should be in an un-intrusive and energy-efficient manner. To save energy switch messages are avoided in favor of a time-out based mechanism. As soon as one node does not receive any switch requests for a certain period of time it switches back to the body monitoring application using the energy mode. Obviously not all nodes fall back at the same time necessarily. But for the benefit of energy savings this temporary inconsistency is tolerated.

Switching control for fast mode and reliable mode
In contrast to body monitoring the health alert needs quick response (running in fast mode) and fire alarm must be set up reliably (running in reliable mode). Therefore the switch to the fast mode and reliable mode is done message based. Broadcast messages were chosen in this scenario to initiate a switch to fast mode and reliable mode. It was assumed that in this restricted scenario broadcasting is a reliable and efficient method to inform all involved nodes. To reduce complexity in this case study multi-hop routing was not considered. Thus, switching to these modes is done identically by broadcasting the switch request repeatedly until the anomalous situation ends.

Conflicting switch requests can only occur for the health alert and fire alarm application (switching to energy mode is done via time out). This conflict is resolved by a higher priority of the reliable mode compared to fast mode. Table 3 summarizes all switching possibilities and the algorithms used to perform the switch.

Table 3. Switching control in case study depending on target mode

Switch from Switch to	**Energy mode**	**Fast mode**	**Reliable mode**
Energy mode		– Event/Message triggered	– Event/Message triggered
Fast mode	– Time triggered		– Event/Message triggered
Reliable mode	– Time triggered	– Not allowed (priority based)	

3.3 Evaluation of the case study

This prototype implementation provided first insights into the advantages and challenges of using the network mode framework.

The use of network modes allowed execution of applications with very different requirements. The network behavior of the small WSN changed according to different conditions and, thus, permitted the coexistence of three applications using the same WSN.

The separation of three network modes has proved advantageous by permitting a modular development and optimization of the network modes. For example, the energy mode was implemented in a first step with a predetermined aggregation node. It was replaced by an adaptive algorithm choosing the most appropriate aggregation node dynamically according to the current situation. The new mode implementation could be applied seamlessly without affecting the function of the other modes.

Any node of a WSN application is able to switch to a different network mode without the need for consistency checks or a central mode control. Consistent switching is performed and controlled by the network mode framework implementation, what eased the implementation of the WSN applications.

Switching between different modes proved to be efficient with respect to the special requirements of the target mode. The switch from energy mode to fast or reliable mode is assumed to be time critical and therefore message based. Thus, the switching time is determined by message transfer time, which resulted in a time of few milliseconds to switch all nodes. Switching to energy mode can take longer in favor of energy savings. The nodes switch within a given time-out (five seconds in the case study) to energy mode without further synchronization or sending messages.

All above mentioned features proved feasible in the small exemplary WSN scenario consisting of eleven sensor nodes. The conceptual advantages still remain true in larger networks and in multi-hop networks. However, the implementations of network modes and the mode switching algorithms have to be adapted in larger networks as discussed below.

4 Related Work and Mode Implementations

We discuss in the following selected approaches for implementing specific modes, which optimize specific goals. Then we consider approaches which introduce flexibility as in our mode concept.

Sensor nodes in WSNs are generally seen as devices with limited resources: limited energy, limited communication capabilities, and limited processing power. Therefore a lot of research work exists addressing to overcome these limitations and to obtain network behavior adapted to these constraints.

Because energy shortage is considered one of the main problems there are many proposals how to organize wireless sensor networks to achieve energy-efficient network behavior. Cluster-based solutions (well-known examples are e.g. LEACH [2], TEEN [4]) build up topologies assigning each node one of the roles cluster-head, gateways, or slave. This can be used to optimize data forwarding and save resources.

Non-hierarchical solutions leverage in-network aggregation considering information on current battery status (SPIN [8]), communication costs on links (MFCA [7]), etc.

Apart from energy-efficiency other aspects like fast real-time communication (SPEED [6]) or reliability [3] are considered, too.

Generally, many solutions exist optimizing *one* mode of network behavior. However, there is no literature how *several* modes can be established and used alternatively in the same WSN. There are proposals how to provide selectable QoS (quality of service) on links or routes ([9]). But there are no solutions how to handle QoS requests not only for dedicated communication paths but for entire distributed WSN applications with a variety of connections. Especially there are no existing frameworks addressing switching between several network modes.

Though the focus of above mentioned literature is not on multiple and selectable network behavior their findings are beneficial for the network mode framework. Protocols and algorithms for energy-efficient, fast, or reliable communication in WSNs can be implemented to serve as one of the network modes. E.g. LEACH, TEEN, etc. are possible candidates for the implementation of energy mode; similarly SPEED can be applied for fast mode and routing protocols providing redundant paths can be used for reliable mode.

Another approach offering a generic procedure to control WSNs is proposed by [5]. In this work roles can be assigned to sensor nodes thereby controlling their function within network protocols, e.g. assigning the roles clusterhead, gateway, or slave in clusters. As claimed in [5], it is also possible to replace whole routing protocols by assigning appropriate roles to the respective nodes. Thus, implementing different protocols in all sensor nodes and switching among them by assigning new roles could be used to achieve different and selectable network behavior as we proposed with our network mode framework. However, the switching itself is not addressed in this work. Thus, in contrast to the network mode framework, switching from one protocol to another would inherently lead to disruptions during switching phase.

5 Conclusion

This paper has introduced the new concept of network modes, which describe different operational modes of a WSN to serve different applications in an optimal way. We argue that there are a few essential modes which are sufficient for a large class of applications which collect data in a sensor network. The modes are energy mode, fast delivery and redundant mode, which offer optimized implementations regarding different requirements. In this way, the applications only specify their needs, but do not need to care about the detailed network operation.

Due to the distributed characteristic of WSNs, setting up and switching between the modes is non-trivial. We have presented a generic mode control layer, which can be used with different mode implementations. To implement this control layer, a new generic framework is proposed which can accommodate different mode implementations in a plug and play fashion. As mode switching can create considerable overhead, a main feature of our framework is to perform switching in different ways depending on requirements the target mode, e.g. energy efficient switching or fast switching. Our framework has to be instantiated with suitable mode implementations and switching options in order to obtain optimized and small software solution for embedded sensor nodes with limited resources.

We have validated our framework in a case study which accommodates different mode implementations and also different ways to switch modes. We have examined the different mode switching options. This includes for instance energy efficient switching using time outs which may lead to temporary inconsistencies. Another issue are conflicts for mode switching, which are handled by mode priorities.

Further work will include more flexible mode control for large networks, which may have different modes running in different or overlapping areas of a network. Also, the flexibility of the modes can be increased by more expressive specification languages to express application needs, see e.g. [11,12]. This could include of in-network processing and more sophisticated data aggregation. Another research issue is the scalability the different mode implementations and switching algorithms with respect to convergence time and overhead.

References

1. J Bosch, Design and use of software architectures: adopting and evolving a product-line approach, - 2000 - ACM Press/Addison-Wesley Publishing Co., New York, NY
2. W. Heinzelmann, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocol for wireless sensor networks, Proc. 33rd Hawaii International Conference System Sciences (HICSS '00), January 2000
3. I. Stojmenovic, Geocasting with Guaranteed Delivery in Sensor Networks, IEEE Wireless Communications, December 2004
4. A. Manfeshwar and D.P. Agarwal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, 1st International Workshop Parallel Distributed Computing Issues Wireless Networks Mobile Computing, April 2001
5. Christian Frank, Kay Römer, Algorithms for Generic Role Assignment in Wireless Sensor Networks, ACM International Conference on Embedded Networked Sensor Systems (Sensys) 2005, San Diego, USA, November 2005
6. J. Kulik, W.R. Heinzelmann, and H. Balakrishnan, Adaptive protocols for information dissemination information in wireless sensor networks, Proc. 5th ADM/IEEE Mobicom Conf. (MobiCom '99), 174-185, Seattle, August 1999
7. F. Ye, A. Chen, S. Liu, and L. Zhang, A scalable solution to minimum cost forwarding in large sensor networks, Proc. 10th International Conference Computer Communication Networks (ICCCN), 304-309, 2001
8. T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. “SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks”, International Conference on Distributed Computing Systems (ICDCS), 2003
9. Mohamed Younis, Kemal Akkaya, Mohamed Eltoweissy, Ashraf Wadaa. "On Handling QoS Traffic in Wireless Sensor Networks," p. 90292a, Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04), Hawaii, 2004
10. K. Romer, O. Kasten, F. Mattern, Middleware Challenges for Wireless Sensor Networks, - ACM SIGMOBILE Mobile Communication and Communications, Review, 2002

11. A. Wolisz, Towards a service definition for (Wireless) Sensor Networks, presentation at ESF Exploratory Workshop on Wireless Sensor Networks, Zurich, 2004, www.vs.inf.ethz.ch/events/esf-wsn04/talks/wolisz.pdf
12. Marco Sgroi, Adam Wolisz, Alberto Sangiovanni-Vincentelli, and Jan Rabaey. A Service-Based Universal Application Interface for Ad-hoc Wireless Sensor Networks (Draft). Unpublished article, November, 2003.