3

Smart Environments and Cross-Layer Design

L. Ozlem KARACA and Radosveta SOKULLU
Dokuz Eylul Univesity, Ege University
Turkey

1. Introduction

In the last decade we have witnessed a really unpredicted boom in the number and variety of applications based on wireless sensor networks (WSN). From environment monitoring and military applications, to health care and event tracking applications, both the diversity and complexity of the nodes themselves and their networked applications have increased immensely (Yick et al., 2008). A combination of consumer demand for more efficient integrated systems and a steep drop in the price of hardware fuelled by manufacturing process improvements has resulted in a noticeable upward cycle of research in the field of networks that not only sense the data but also provide automated reaction to specific situations known as Wireless Sensor and Actuator Networks (WSAN) (Akyildiz & Kasimoglu, 2004). "Smart environments" are discussed as the next step in these evolutionary developments in intelligent systems automation related to utilities, construction, industry, home and transportation. The "smart environment" is defined as one that is "able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment".

The WSN, which are in the heart of the "smart environments" consist of densely deployed microsensor nodes that continuously observe certain physical phenomenon. The existing abundance of WSN applications can be divided into two major groups based on the nature of the supported applications: WSN for monitoring and WSN for event detection/tracking. A major common feature is that both exploit the collective effort of nodes which have computing, transmitting and sensing capabilities. From the user point of view the main objective of WSN is to reliably detect or collect, and estimate event features based on the collective information provided by all sensor nodes. From the engineering design point of view, the main challenge for achieving this objective is posed by the severe energy and processing constraints of the low-end wireless sensor nodes. The collaborative sensing notion of WSN, which is achieved by the networked deployment of sensor nodes, can potentially be used towards overcoming the characteristic challenge of WSN, i.e., resource constraints. To this end, there has been a significant amount of research effort to develop suitable networking protocols in order to achieve communication with maximum energy efficiency. Because of the strict demands of WSN as compared to wired networks and Ad-Hoc networks, the design goals of such system are different from the traditional approaches. The suitability of one of the foundations of networking, the OSI layered protocol architecture, is coming under close scrutiny from the research community. It is repeatedly argued that although layered architectures have served well for wired networks, they are not particularly suitable for wireless sensor networks. That is why the notion for a different approach, called cross-layer design, has come into existence.

Generally speaking, cross-layer design refers to protocol design done by actively exploiting the dependence between protocol layers to obtain performance gains. This is unlike layering, where the protocols at the different layers are designed independently (Srivastava & Motani, 2005). Cross-layer design stands as the most promising alternative to inefficient traditional layered protocol architectures allowing researchers to take into consideration different factors like the scarce energy and processing resources of WSNs, joint optimization and design of networking layers and last but not least overall performance evaluation. Accordingly, an increasing number of recent papers have focused on the cross-layer development of wireless sensor network protocols (Melodia et al., 2006). Recent papers (Cui et al., 2005); (Fang & McDonald, 2004); (van Hoesel et al., 2004); (Vuran et al., 2005) reveal that active cross-layer interactions and integration incorporated in the design techniques can bring about significant improvement in terms of energy conservation. The reasons have been summarized as follows:

- The significant overhead of layered protocols results in high inefficiency.
- Recent empirical studies necessitate that the properties of low power radio transceivers and the varying wireless channel conditions should be included in the protocol design.
- The severe restrictions on capabilities such as storage, processing and especially energy of the wireless sensor nodes make active interaction between different protocol layers mandatory.
- The event-centric approach of WSNs requires application-aware communication protocols.

It is obvious that the necessity has emerged for creating a new model that will inherently take into consideration the abovementioned specifics and restrictions of WSN.

Examining the literature in the area of cross-layer design, the following important observations can be made (Srivastava & Motani, 2005). First, there are several interpretations of cross-layer design. This is probably because the cross-layer design effort has been made rather independently by researchers from different backgrounds, who work on different layers of the stack. Second, some cross-layer design proposals build upon other cross-layer designs, hence some more fundamental issues (coexistence of different cross-layer design proposals, when cross-layer design proposals should be invoked, what roles the layers should play, etc.) are not addressed directly. Third, the question of how cross-layer interactions may be implemented has not been examined sufficiently; therefore the relation between the performance viewpoint and implementation concerns is weak. Furthermore, the wireless medium allows richer modalities of communication than wired networks. For example, nodes can make use of the inherent broadcast nature of the wireless medium and cooperate with each other. Employing modalities like node cooperation in protocol design also calls for cross-layer design.

Another very important aspect is related to the realization of the idea - cross-layer design proposals realized by different ways and manner exist in literature. Some of them focus on the idea of how actions in one layer affect other layer or layers (Wang & Abu-Rgheff, 2003); (Sichitiu, 2004). Studies exist also that consider the combined actions in two or three layers (Melodia et al., 2006); (Akyildiz et al., 2006); (Lee, 2006). However a cross-layer solution generally decreases the level of modularity, which may lead to decoupling between design and development process, making it more difficult to design further improvements or introduce innovations. Moreover, it increases the risk of instability that can be caused by unintended functional dependencies, which are not easily foreseen in a non-layered architecture. Issues like these should be especially considered when trying to create and overall model or framework reflecting the inherent features and requirements of WSN.

Although a consistent amount of recent papers have focused on cross-layer design and improvement of protocols for WSNs, a systematic methodology to accurately model and leverage cross-layer interactions is still missing. Furthermore, the definition of a suitable, encompassing both performance and implementations issues cross-layer design (CLD) framework is required to unify the abundant research in WSN. Towards this aim we investigate the few suggested so far proposals for CLD frameworks which have quite different features and implementation methods focusing on the performance improvement and the consequent risks of a cross-layer design approach.

In this chapter we first introduce the cross-layer protocol design methodology for WSN and WSAN and review some major sources in literature. We focus on the concept of CLD frameworks, as a new emerging approach contrasting the well known conventional layered approach of protocol design. Our first aim is to investigate the ongoing work in the area of CLD framework, put that work in perspective, and consolidate the existing results and insights. Our second aim is to define some major criteria for comparing such frameworks and identify their pros and cons in terms of adaptivity, power efficiency, complexity, channel property orientation and fault tolerance.

From here on the chapter is organized as follows. In Section 2 we overview the concept of cross-layer design and the necessity for the development of CLD frameworks. In Section 3 we provide a definition of CLD framework and present a brief survey of the existing CLD frameworks in literature. Further elaborating on that subject in Section 4 we propose a set of criteria relevant to the evaluation of CLD frameworks and provide a detailed comparison of the discussed frameworks. Finally in Section 5 we provide a look ahead by discussing WSAN and the protocol design issues they pose. The chapter is concluded with some open research issues that we foresee for the development of a unified approach to protocol design in sensor networks suitable for smart environments.

2. Cross-Layer Design and Frameworks

To understand the concept of the cross-layer design and CLD frameworks, first the definition of layered frameworks should be elaborated. A layered architecture, like the seven-layer open systems interconnect (OSI) model (Stallings, 2006), divides the overall networking task into layers and defines a hierarchy of services to be provided by the individual layers. The services at the layers are realized by designing protocols for the different layers. The architecture restricts direct communication between nonadjacent layers; communication between adjacent layers is limited to procedure calls and responses.

Alternatively, protocols can be designed by violating the reference architecture, for example, by allowing direct active information exchange between protocols at nonadjacent layers or sharing variables between layers. Such violation of the layered architecture is what is known as the most popular definition of cross-layer design with respect to the reference architecture (Srivastava & Motani, 2005). There exist a number of studies that discuss and evaluate the cross-layer design approach from different angles and formulate different positions on its applicability and possible disadvantages (Srivastava & Motani, 2005); (Melodia et al., 2006); (Zhang & Zhang, 2008); (Raisinghani & Iyer, 2004); (Wang & Abu-Rgheff, 2003); (Zhang & Cheng, 2003). However, the work of Srivastava and Montani (Srivastava & Motani, 2005), stands out as one of the most completed classifications available. The article presents detailed definitions and classification of cross-layer design and related interlayer interactions and the authors dutifully argue that they present a "taxonomy for classifying the existing cross-layer proposals and clarify the different interpretations of cross-layer design". Fig.1 summarizes their suggested taxonomy. They classify the possible methods for realizing cross-layer design in 6 groups and present examples for each one. The suggested taxonomy takes into consideration the interlayer interactions and their direction as well as the possible merging of layers up to the point where a totally holistic structure can be achieved (called "vertical calibration").

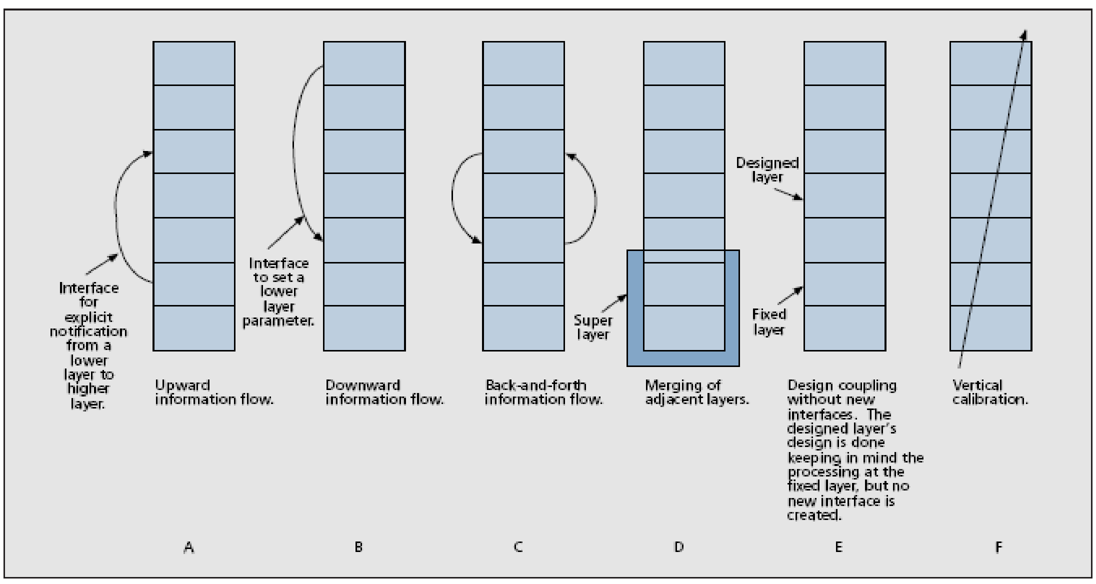


Fig. 1. Illustrating the different kinds of cross-layer design proposals. The rectangular boxes represent the protocol layers (Srivastava & Motani, 2005).

Another considerable attempt to put the discussion on cross-layer design on a well structured ground is given in (Melodia et al., 2006). The authors suggest a systematic methodology to model and leverage cross-layer interaction based on the assumption that the design of networking protocols for multi-hop sensor networks can be interpreted as the joint solution of resource allocation problems at different protocol layers. Thus they classify the proposals available in literature based on the number of protocol layers involved and the layers in the classical OSI model they try to replace. The focus is on expected performance improvement and the risks involved in the cross-layer approach. It is clearly stated that cross-layer solutions decrease the level of modularity and significantly increase the risk of instability brought by unforeseen functional dependencies and a joint solution is required.
(Zhang & Zhang, 2008) stress on the fact that cross-layer design allows active communication between different layers which ultimately can result in significant performance gains. Some of the new trends in wireless networking such as cooperative communication and networking, opportunistic transmission and real system performance

evaluation are discussed in light of QoS support for multihop sensor networks. The interaction between protocols at different layers is examined from the point of view of different system parameters controlled at distinct layers. For instance, it is argued that power control and modulation adaptation in the physical layer can affect the overall system topology, while scheduling and channel management in the MAC layer will affect the space/time reuse in the whole network. By using a general framework (Fig.2) they illustrate the interaction ideas and point out that all controls can have a multiple impact. (1) in Fig.2 illustrates the fact that assignment of channels to certain network interfaces changes the interference between neighboring channels. The authors conclude by pointing out that in order to achieve joint optimization of the whole system it is absolutely necessary to consider that all controls do cross different layers.

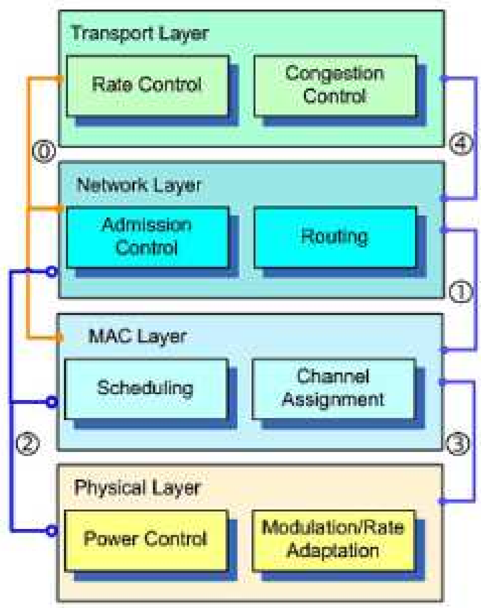


Fig. 2. Cross-layer framework and interaction among layers (Zhang & Zhang, 2008).

The experience gained through both scientific studies and experimental work in WSNs revealed important interactions between different layers of the network stack. These interactions are especially important for the design of communication protocols for WSNs.

The purpose of design principles is to organize and guide the placement of functions within a system. Design principles impose a structure on the design space, rather than solving a particular design problem. This structure provides a basis for discussion and analysis of trade-offs, and suggests a strong rationale to justify design choices. The arguments would also reflect implicit assumptions about technology options, technology evolution trends and relative cost tradeoffs. The architectural principles therefore aim to provide a framework for creating cooperation and standards, as a small "spanning set" of rules that generates a large, varied and evolving space of technology (Carpenter, 1996).

The general description of a framework states that it is a “basic conceptual structure” used to solve or address complex issues. A framework can be defined as an extensible structure for describing a set of concepts, methods and technologies necessary for a complete product design and manufacturing process. Regarding the CLD framework we can say that it should incorporate and reflect the inherent characteristics and specifics of WSN, and address the

major issues of performance and implementation in a joint manner for providing enhanced operation, energy efficiency and extending the lifetime of the network. As discussed before, numerous cross-layer solutions have been proposed so far taking into consideration a single or only a few, (mostly a combination of two or three) of the parameters of the WSN. Unfortunately the changes made affect other layers and might give rise to totally unpredicted situations and problems. Even if these situations and problems do not arise every time, in a different application, the suggested approach most probably will not provide the same functionality and optimization (Kawadia & Kumar, 2005); (Shakkottai et al., 2003); (Zhao & Sun, 2007).
To summarize, it is important to consider and evaluate the suggested cross-layer approaches in light of a basic conceptual structure, which is independent of the specific application and can provide adaptivity to system changes. In the next section, we continue with a survey, discussion and evaluation of the CLD frameworks suggested by different researcher teams.

3. Cross-Layer Design (CLD) Framework Proposals

To achieve understanding of WSN protocol design in terms of constituting CLD frameworks, we investigate four different CLD framework proposals. We examine each of them, in this section and give details of these proposals and their main features.

3.1 TinyCubus

Known applications of WSN fall into different classes and based on this the possible approaches to building a CLD framework can be subdivided into two major groups. The first one is using generic components and definitions while the second is using several more specific components or entities for each different class of applications. In (Marrón et al., 2005a) the architecture of a generic framework is presented, since its internal structure is the same independently of whether or not it is intended for all classes or just a certain number of applications.
The architecture of TinyCubus presents a single generic framework that can support very different application requirements even with contradictory requirements like environmental monitoring or target tracking. Its aim is to provide the necessary infrastructure to support the complexity of a specific WSN system architecture. TinyCubus consists of a Data Management Framework, (DMF) a Cross-Layer Framework, (CLF) and a Configuration Engine (CE). (Marrón et al., 2005b) The Data Management Framework allows the dynamic selection and adaptation of system and data management components. The Cross-Layer Framework supports data sharing and other forms of interaction between components in order to achieve cross-layer optimizations. The Configuration Engine allows code to be distributed reliably and efficiently by taking into account the topology of sensors and their specific assigned functionality.
The overall architecture of TinyCubus mirrors the requirements imposed by the two applications namely CarTalk 2000 (Tian & Coletti, 2003); (Morsink et al., 2003) and Sustainable Bridges (Marrón et al., 2005c) and the underlying hardware. It has been developed with the goal of creating a totally generic and fully reconfigurable framework for sensor networks. As shown in Fig. 3, TinyCubus is implemented on top of TinyOS using the nesC programming language, which allows for the definition of components that contain functionality and algorithms. The applications register their requirements and components with TinyCubus and are executed by the framework.

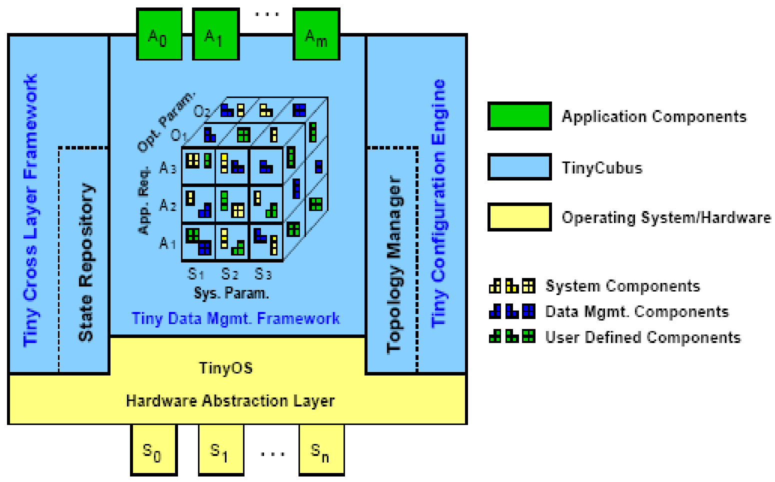


Fig. 3. Architectural components in TinyCubus (Marrón et al., 2005b).

The major design goal of TinyCubus is to support different application schemes easily and to do so it uses a generic framework. Despite all the differences, many applications obviously have some commonalities. Therefore, it is possible to simplify the development of both applications – and of others that share some properties with them.

Below the three major components of the TinyCubus Framework are discussed in more detail:

1. **Tiny Cross-Layer Framework**: The goal of the Tiny Cross-Layer Framework is to provide a generic interface to support parameterization of components using cross-layer interactions. The Tiny Cross-Layer Framework provides support for both parameter definition and custom code execution. This framework uses a specification language that allows for the description of the data types and information required and provided by each component. This cross-layer data is stored in the state repository. To deal with custom code, the cross-layer framework makes use of TinyCubus' ability to execute dynamically loaded code.
    a. State Repository: The cross-layer framework acts as a mediator between components. Cross-layer data is not directly accessed from other components but stored in the state repository. Thus, if a component is replaced (e. g., to adapt to changing requirements), no component that uses the old component's cross-layer data is affected by the change, given that the new component also provides the same or compatible data.
    b. Custom Code: The approach used in this study does not extend the interface of all components between two interacting ones. Instead, it provides support for the execution of application-specific code in lower-layer components via callbacks.
2. **Tiny Configuration Engine**: The Tiny Configuration Engine makes possible installation of new components, or swapping certain functions if necessary, by distributing and installing code in the network. Its goal is to support the configuration of both system and application components using cross-layer information about the functionality assigned to the nodes.
    a. Topology Manager: The topology manager is responsible for the self-configuration of the network and the assignment of specific roles to each node. A role defines the function of a node based on properties such as hardware capabilities, network neighborhood, location etc. Examples for

roles are SOURCE, AGGREGATOR, and SINK for aggregation, CLUSTERHEAD, GATE- WAY, and SLAVE for clustering applications as well as VIBRATION to describe the sensing capabilities of a node.

b. Code Distribution: Most existing approaches that distribute code in sensor networks do it by replacing the complete code image. However, most of the time only a single component needs to be updated or replaced. To avoid wasting energy by sending complete code image, configuration engine only transmits the components that have changed and integrates them with the existing code. The code distribution depends on the role of the node. Code updates only send to those nodes that belong to a given role and need this code update.

3. **Tiny Data Management Framework**: The goal of the Tiny Data Management Framework is to provide a set of standard data management and system components and to choose the best set of components based on three dimensions, namely system parameters, application requirements, and optimization parameters. The cube of Fig.1, called 'Cubus', represents the conceptual management structure of the Tiny Data Management Framework. When developing a suitable algorithm, at first, influencing factors called system parameters, such as density or mobility of the network is considered. Secondly, application requirements, such as reliability requirements, additionally restrict the set of possible algorithms. Finally, the algorithm is selected that fulfills best some optimization criteria, e. g., minimal energy consumption.

The strongest point in this framework proposal is its high adaptivity, the fact that it can be used for a number of different classes of applications. However, this comes at the price of high complexity and very general consideration of the wireless medium modalities.

3.2 DMA-CLD and the Optimization Agent Based Framework

The Optimization Agent Based (OAB) Framework (Lee, 2006) which is an extension of the cross-layer interaction approach suggested as the Dynamic Multi-Attribute Cross-Layer Design (DMA-CLD) constitutes a different class of framework for WSNs. It is based on the idea of systematically organizing the interactions between the layers by means of defining an optimization agent, serving as a core repository or database where essential information is maintained temporarily and exchanged across the protocol stack.

The DMA-CLD approach (Safwat, 2004), is proposed for cross-layer interactions in wireless ad-hoc and sensor networks to allow multiple, and possibly conflicting (single-layer, cross-layer, nodal, and networking) objectives to be met concurrently. While preserving the OSI layered structure, DMA-CLD allows interactions both upwards and downwards in the stack, i.e. information from the network layer can be passed both to higher or lower layers like the application and the MAC layers. It utilizes the Analytic Hierarchy Process (AHP) for making multiple, and possibly conflicting decisions. Thus the DMA-CLD can be viewed as a multi-objective framework that can be extended to accommodate any number of objectives and can relate to any number of OSI layers. It considers the network as a whole and reflects the objectives of selected "best network performance" on the parameters of the single node. DMA-CLD framework accepts a set of routes in the network, which are chosen to optimize the network performance according a given criteria ("high remaining battery capacity", "reliable packet delivery", etc.), as input.

The main idea of DMA-CLD is presented in Fig. 4.

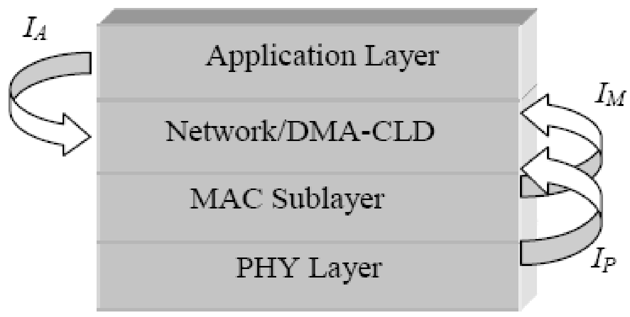


Fig. 4. The DMA-CLD framework and the associated cross-layer interactions (Safwat, 2004).

The key point involved in this approach is choosing multiple routes depending on a comparison matrix which includes the objectives listed precedence. It alleviates congestion by using multiple routes. The routes are ranked according to the Analytic Hierarchy Process (AHP). Putting together the information passed from the application, MAC and PHY layer a reciprocal pairwise comparison matrix C = [ci, j] is constructed for the multiple attributes (equation 1).

$$ci, j = \frac{1}{cj, i}, \forall i, j \in \Omega \qquad (1)$$

where $\Omega \neq \varphi$ is the set of objectives. DMA-CLD computes a priority eigenvector via which each objective is assigned a priority. The eigenvector indicates how well each route satisfies each objective. The system also considers route outage. It is calculated by:

$$P_O = 1 - e^{-\frac{\gamma_T}{\bar{y}}} \qquad (2)$$

where P_o is the link outage probability when the SNR threshold is γ_T and the average SNR is γ. The "route outage" value can be used by inter-layer feedback mechanism on the PHY layer. Thus, the operation of the DMA-CLD approach can be summarized as follows:

- The DMA-CLD is executed at the network layer. There the routes are ranked based on inter-layer feedback (provided by the interfaces I_A, I_M, I_P) and information from intermediate nodes and the first M paths are used for simultaneous load-balanced routing.
- The I_M interface is in charge of relaying MAC-specific information, such as the number of one-hop neighbors and the contention index, to the network layer.
- Information pertaining to the physical layer and the channel conditions, which is reflected in calculating the route outage, is carried to the network layer via the I_P interface.
- The application layer dynamically constructs the "pairwise attribute comparison matrix" taking into account the application requirements and network conditions such as traffic type, transmission delay bound, and transmission delay jitter bound. Then the reciprocal matrix C is constructed and conveyed to the network layer via the I_A interface.

The ideas involved in DMA-CLD were further extended in the OAB Framework, presented in (Lee, 2006). The major contribution of OAB is combining the inter-layer interactions as described in DMA-CLD in the form of a core repository, namely Optimization agent. The structure of the suggested framework is given in Fig. 5.

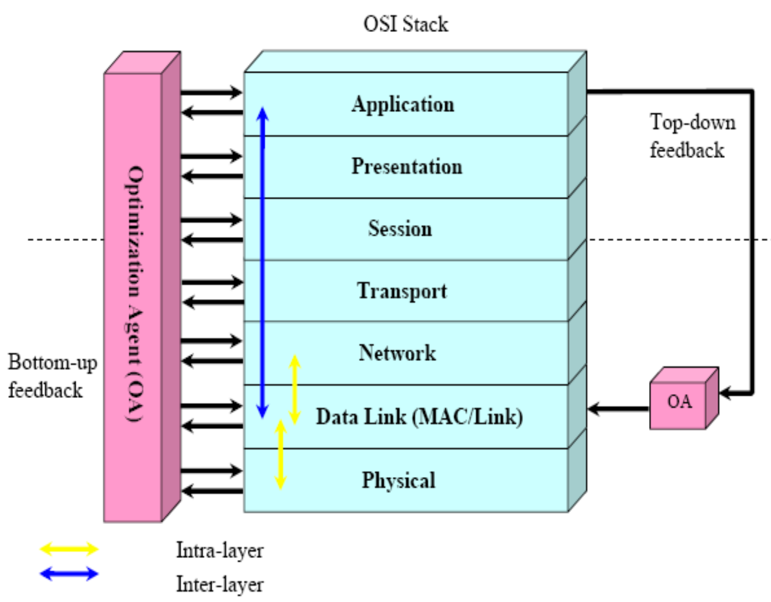


Fig. 5. The interactions of layers in Optimization Agent based design (Lee, 2006).

In the OAB framework the authors categorize the interactions between layers in two general groups: intra-layer (between adjacent layers) or inter-layer interactions (across two or more adjacent/nonadjacent layers). Both can be executed bottom up or top down.

- Bottom up interactions represent the typical feedback mechanism used in control systems. For example, information about the channel conditions at the physical layer is used at the link layer to adapt its error control mechanisms or at the application layer to adapt its sending rate.
- Top down interactions can be described as sending messages for the normal operation or data flow. An example is the sending of urgent messages for prioritized traffic from the application layer to the network layer or sending information from the MAC layer for tuning the transmission range at the PHY layer.

The structure of the OAB provides a framework that can accommodate changes or modifications to the protocol stacks for different network requirements or applications. It presents a generalization of a number of approaches that intend to optimize the performance between adjacent layers (e.g. MAC and network layers) (Liu et al., 2004); (Alonso et al., 2003). It extends the cross-layering process to all protocol layers as critical information kept in the OA can be exchanged across all layers and thus the performance is jointly optimized.

When compared to other frameworks the DMA-CLD and its extension OAB framework provide a direct possibility to take into consideration both channel oriented parameters and power efficiency by defining suitable objectives that influence the decision at the network layer. However the selection of the inputs for the reciprocal pairwise matrix is a very sensitive issue and the involved computational resources are considerable as the decisions have to be taken in real time. Also the mechanism of accessing the information in the suggested OA and possible concurrency issues or race conditions have to be further elaborated as they pose a potential pitfall.

3.3 Horizontal Framework

In their work (Hakala & Tikkakoski, 2006), the authors suggest reducing the size and functions of the protocol stack and propose an additional cross-layer management entity to make application programming easier by simplifying the protocol stack in a way to better suit the limited resources available in WSNs. The role of the cross-layer management entity in this study is to offer a shared data structure and to take care of sensor network specific functions, like topology management and power saving. It also provides additional services that applications and other layers in the protocol stack can use. Data structures, which are in common use, are also implemented in the cross-layer management entity. So the two major entities, Application and Protocol Stack are responsible for the application-specific data transmission.

The cross-layer implementation provides reduced computational and memory requirements - not all the information needs to be transmitted between application interfaces and protocol layers. The other advantage is that the architecture also allows the implementation of the application and protocol stacks to be as simple as possible, since they are practically free of the tasks related to network management.

While taking into consideration some of the sensor network's special needs, it is obvious that there is a necessity of different solutions to be used. The system proposed uses horizontal architecture instead of the vertical model. Fig. 6 illustrates the major idea and components of the suggested horizontal CL framework for WSNs Above the physical layer and data link layer which are kept like in the classical structure, the architecture branches into two parallel areas. The ***Application*** and the ***Protocol Stack*** are responsible for the application-specific data transmission and the ***Cross-Layer Management*** (CLM) ***Entity*** takes care of network management. The CLM Entity is further divided into two parts: Management Entity, and Shared Data Structures.

The Management Entity is made up of one or more parallel modules, each of which takes care of a task affecting the operation of the sensor network node. Examples of these tasks include network management based on listening beacon messages, implementing a control algorithm that improves power saving characteristics, selecting efficient data transmission routes and so on.

The CLM entity is responsible for tasks directly related to the operation of the network but also general purpose tasks that are common to most WSN applications. Some of these, representing important modules in the CLM entity are summarized below:

- **Network configuring and topology management** –Topology management is an important cross-layer issue that is included in the CLM entity. It is vital to monitor the state of the surrounding network, for example, battery charges in neighboring nodes, network control traffic including beacon messages or other control messages. Using the information provided by the CLM entity, resources of the network can be employed effectively.

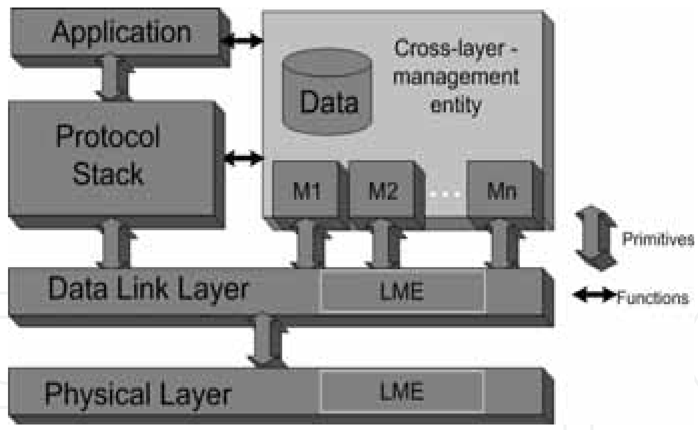


Fig. 6. Horizontal cross-layer architecture (Hakala & Tikkakoski, 2006).

- **Providing optimal data transmission routes:** Routing in the WSN is a major factor in providing efficient network operation. In a lot of cases multi-hop and more power efficient methods might be sought then the general flooding algorithm. Deciding in the optimal route affects both the operation of the single node and its duty cycle and the topology of the whole network so it is considered one of the main modules in the CLM entity.
- **Providing optimal power mode selection for the node:** This includes tasks as moving the node into power saving mode or providing other power related solutions whenever feasible:
  - For the implementation of short duty cycles, the mechanism such as on/off type switching can be used. To extend the lifetime of a battery-powered device into many years, the duty cycle must be as short as possible.
  - Selection of the node's optimal transmitting power is also classified as a power saving issue. Listening consumes more energy than sending, because the receiver must be kept on independent of whether there is any traffic on the channel or not. However, energy can be saved by adjusting the transmitter power. This also provides that disturbances to other nodes are minimized.
- **Sharing data structures:** Lot of the operations in the network as self-configuration, routing information exchange, power saving etc. are interrelated. For this reason they cannot be easily confined to any particular layer. To minimize memory and computational requirements, the authors suggest the use of the so called Shared Data Structures. An example of such usage is adjusting the optimal broadcast power knowing the neighbor's data. However, Sharded Data Structures have to be very clearly defined as there might be unforeseen dependencies.
- **Coding/decoding:** Coding/decoding is a general purpose operation is not dependent on the protocol stack used. Therefore, it can be done in the CLM entity. Algorithms used in coding may include, among others, different compression and encryption algorithms.

As can be deducted from the discussion presented above the main idea of the Horizontal Framework is to simplify the protocol stack and separate certain tasks as modules of the CLM entity, thus making application programming easier. The low stack reduces the data transfer between the different layers. At the same time, the reduced header information by means of the CLM entity results in a reduced number of bits to be transmitted. Power consumption in data transmission is directly proportional to the length of the broadcasted frame, so the system ensures extending network lifetime. The interface between the CLM entity and the Application/Protocol Stack employs the client/service principle. The CLM entity can provide certain services that the layers in the protocol stack and the application can use. Usually, the function of communication in this interface is to perform a certain task, for example the updating of Shared Data Structures. Because the application program can be freed from the tasks related to network management and some general purpose tasks, it is possible to have a very simple application program. The system also allows the use of the same sensor network structure for a great number of different applications.

The Horizontal framework provides high degree of adaptivity to different applications while at the same time involves much less complexity then the TinyCubus framework. The suggested management entity directly interacts with the MAC layer, with the network and application layer providing duty cycle control, topology control and other solutions to extent the overall lifetime of the network. However it does not define how modifications in the Shared Data Structures should be taken into account. The dependencies between the modules and the suggested common data structures might bring out unexpected complicacy. In the example presented by the authors, two management modules are proposed – the power saving and the topology control module. They do provide the required efficiency related to the example at hand (CiNet) but for other applications the number of these modules might have to be increased resulting in a much higher complexity.

3.4 XLM

XLM (cross-layer module) (Akyildiz et al., 2006) is a unified cross-layer module which is developed to achieve efficient and reliable event communication in WSNs with minimum energy expenditure. XLM merges common protocol layer functionalities into a single cross-layer module for resource-constrained sensor nodes. The operation of the XLM is devised based on a new notion, which the authors define as "initiative determination". It is the core of the XLM and implicitly incorporates most of the the inherent communication functionalities required for the successful operation of a general application oriented WSN. Based on the initiative concept, XLM performs received based contention, local congestion control, and distributed duty cycle operation in order to realize efficient and reliable communication in WSN.

The basis of communication in XLM is built on initiative concept. In this concept, each node decides whether join a network and participate a communication or not according to the initiative value. Consequently, a completely distributed and adaptive operation is deployed. The next-hop in each communication is not determined in advance. Instead, an initiative determination procedure is used for each node to decide on participating in the communication.

Operation based on the initiative concept in (Akyildiz et al., 2006) can be summarized as follows: A node starts transmission by broadcasting an RTS packet to indicate its neighbors that it has a packet to send. Upon receiving an RTS packet, each neighbor of node **i** decide to participate in the communication or not. This decision is given through initiative determination. The initiative determination is a binary operation where a node decides to participate in communication if its initiative is 1. Denoting the initiative as I, it is determined as follows:

$$I = \begin{cases} 1, if \begin{cases} \xi_{RTS} \geq \xi_{Th} \\ \lambda_{relay} \leq \lambda_{relay}^{Th} \\ \beta \leq \beta^{\max} \\ E_{rem} \geq E_{rem}^{\min} \end{cases} \\ 0, otherwise \end{cases} \quad (3)$$

The initiative determination value is calculated based on four variables. Each of them represents a necessary threshold value that should be satisfied. The initiative is set to 1 if all four conditions declared above are satisfied. Each condition in inequality (3) constitutes certain communication functionality. The first condition ensures that reliable links are to be constructed and for this purpose, it requires that the received signal to noise ratio (SNR) of an RTS packet, ξ_{RTS}, is above some threshold ξ_{Th} for a node to participate in the communication. The second and third conditions are used for local congestion control. The second condition prevents congestion by limiting the traffic a node can relay. The third condition ensures that the node does not experience any buffer overflow and hence, also prevents congestion. The last condition ensures that the remaining energy of a node E_{rem} stays above a minimum value, $E_{min\ rem}$. This constraint guarantees even distribution of energy consumption. The cross-layer functionalities of XLM are summarized in these constraints defining the initiative of a node to participate in communication.

Each node performs distributed duty cycle operation. The value of the duty cycle is denoted by δ and defines the ratio of the time a node is active. Each node is implemented with a sleep frame with length TS sec. As a result, a node is active for δ × TS sec and sleeps for (1 − δ) × TS sec. There are two main duties according to which sensor nodes can be classified: source duty and router duty. The source duty refers to the nodes with event information that need to transmit their packets to the sink; hence these types of nodes can select their rates based on the congestion in the network. The router duty refers to the nodes that forward the packets received from other nodes to the next destination. These nodes indicate their initiative on accepting new flows through their path to the destination. Based on these duties, each node determines its initiative to participate in the transmission of an event as explained above.

When a node wants to send a packet, it first listens to the channel. If the channel is idle, the node broadcasts an RTS packet, which contains the location of the sensor node i and the location of the sink. By getting the packet, other nodes in networks, decide whether or not they are located in a feasible region or in an infeasible region. The node located nearer to sink is "in feasible region", otherwise it is "in infeasible region". Only nodes located in feasible region initiate the procedure, nodes located far are switched to sleep mode to save energy. If a node decides to participate in the communication, it performs receiver contention. Following the receiver contention procedure node i receive a CTS packet from a potential receiver and send a DATA packet indicating the position of the winner node in the header so the other nodes stop contending and switch to sleep. Since each time only a small number of nodes contend in the selected "priority regions" the collision probability is small in XLM.

Two sources of traffic are considered as an input to the buffer of each node:

- Generated packets: The sensing unit of a node senses the event and generates the data packets to be transmitted by the sensor node during its source duty. It is referred to these packets as the generated packets. For a node i, the rate of the generated packets is denoted by λ_{ii}.
- Relay packets: As a part of its router duty, a node also receives packets from its neighbors to forward to the sink due to multi-hop nature of sensor networks. These packets are referred as the relay packets. The rate at which a node i receives relay packets from a node j is denoted as λ_{ji}.

The main idea of XLM cross-layer congestion control is to regulate the congestion. XLM has two main congestion control measures:

- In router duty - enabling the sensor node to decide whether or not to participate in the forwarding of the relay packets based on its current load arising from its relaying functionality
- In source duty - explicitly controlling the rate of the generated data packets.

For realizing congestion control, besides regulating the relaying functionality by the initiative determination, the XLM allows local congestion control by directly regulating the amount of traffic generated and injected to the network at each node.

This framework presents a novel approach in considering a number of network and physical layer requirements by combining them in a very simple structure. However it does not include any fault tolerant mechanisms and being predominantly a network layer based solution does not directly address any issues at the application layer. It also implicitly assumes that all nodes have exact information about their own location and centralized information about the location of the sink.

After this overview of the suggested in literature examples of CLD Frameworks, we proceed, in the next section with a discussion of the relation between WSN application requirements and the functionality of a basic conceptual protocol structure that would meet the specifics and limitations of WSN protocol design.

4. Evaluation of the Existing Frameworks

After suggesting a possible unified approach to comparing diverse WSN application, the Application Comparison Matrix, in the section above, our discussion continues with an attempt to define suitable criteria for evaluating CLD frameworks. Further on in this section we propose a detailed comparison of the CLD frameworks surveyed in section 3.

- **Adaptivity:** The adaptivity evaluates the extent to which a framework can easily and in a fine grain manner adapt itself to the changes in the requirements of heterogeneous applications, to different hardware platforms and to different network topologies. As can be seen from the selected applications, sometimes the differences in their requirements can be even conflicting. For example the Sustainable Bridges application (Marrón et al., 2005a; 2005b; 2005c; 2005d) implies a pushed based data model while the Car Talk 2000 (Tian & Coletti, 2003; Morsink et al., 2003) needs a pull based one. In some very specific oriented applications, like for example Forest Fire Detection (CRUISE 2007) nodes might perform very simple tasks and the required hardware might be greatly simplified, while in others like Sense-R-Us (Lachenmann et al., 2005) the need for diverse information collection and its management might require more sophisticated hardware platforms and functionality. Last but not least changes can occur because of the highly erratic nature of the wireless channel which reflects directly on the network topology and connectivity.

- **Power efficiency:** The most restricted resource in wireless sensor networks is the power of the nodes. It is very important how the suggested framework takes this issue into account. In some frameworks like for example the XLM the power efficiency is considered in a totally distributed manner, at the single node level. On the other hand in the Horizontal Framework this issue is considered both at the node level, by introducing a special management module called the "power saving module" and at the network level by the so called "topology control module". Thus by introducing different modules, the Horizontal Framework provides possibilities for versatile and fine grained control over the power consumption in the node iteself and in the network as a whole. In this respect the TinyCubus provides the most detailed approach but of course at the price of very high complexity.
- **Channel-oriented:** Wireless channel is inherently unsteady. The frameworks that take into consideration this feature can be classified as channel-oriented. They allow for fine tuning of the network operation and management involving in a fairly direct way the channel characteristics.
- **Fault tolerance:** There are many sources that might alter the successful transmission of information and the efficient operation of the network as a whole. Faults might originate because of the mobility of the nodes, fluctuations of the channel, excessive channel utilization due to high density deployments etc. Measures should be taken to minimize the effect of such phenomena and their effect on the network. The fault tolerance criterion takes into account how such issues are covered in the suggested framework.
- **Complexity:** A proposed framework might take into consideration all possible cases and specifics related to a large number of applications but this would result in a structure too difficult to implement and manage. The complexity is an important implementation oriented parameter that has to be taken into account when evaluating the CLD framework.

The design goals and main concerns of the frameworks discusses above are quite different and each has distinctive features, advantages and disadvantages from a specific point of view. Based on the criteria specified we classified the existing frameworks and the results are presented in the Table 1. below:

Property	**TinyCubus**	**DMA-CLD**	**Horizontal**	**XLM**
Adaptivity	■■■■	■■	■■	■
Channel-oriented	■■■■	■■	■■■	■■■■
Power efficiency	■■	■■	■■■■	■■■■
Fault tolerance	■■	■■	■■	■
Complexity	■■■■	■■■	■■	□

□ Not important ■Little ■■ Medium ■■■High ■■■■ Very important

Table 1. Frameworks comparison table.

TinyCubus aims to provide a framework that can easily and in a fine grain manner adapt itself to the changes arising from heterogeneous applications, to different hardware and to different network operation. The topology manager in the TinyCubus framework and the role-based code distribution algorithm are used to provide dynamic code distribution and allow very high degree of adaptivity. This framework can be applied quite successfully to develop both applications like Sustainable Bridges and Forest Fire Detection as well as more complex interaction-based ones like the Sense-R-U and CarTalk 2000. In (Marrón et al., 2005a) it is proven that the role-based code distribution algorithm reduces the messages sent to nodes which need update information compared to general flooding. Suitably selected algorithms can be applied for regulating the duty cycle for sending and receiving mode allowing medium to high degree of energy efficiency. Also, mobility of the nodes and partially the specifics of the transmission channel/environment can be taken into consideration by distributing suitable code using the CE. Even though not explicitly mentioned in the article, with some further effort, fault tolerance issues can be incorporated. However, on the other hand, the TinyCubus, being so detailed and encompassing, is far more complex when compared to other frameworks. From implementation point of view it presents a real challenge. The complexity evaluation based on the number of messages to be exchanged for distributing new code relies on a single and very restricted example which does not justify the general case.

The DMA-CLD and also the OAB frameworks present an interesting view for creating a "common entity" used to simplify the traditional protocol stack and provide more efficient network operation. It builds on the general direction of the research in CL design and optimization so far that evolves around inter-layer and intra-layer interactions and parameter exchange. The functions of the existing layers are kept intact, while the data structures and available data are unified in a common entity. Thus it can provide high degree of channel-oriented operation because the common access to data about the channel conditions can be used directly by other layers to optimize performance at node and network level. Also certain degree of interoperability will be ensured as the layered stack is preserved. Even though existing work in CL design based on optimization of the operation of two or more layers, proves that such type of solutions do bring overall energy efficiency the suggested approach has some pitfalls. First of all, the access to the OA is a potential source of problems and can bring about additional complexity instead of reducing complexity. Second, race conditions will be difficult to track and deal with. Last but not least the suggested approach does not allow for efficient and adequate to WSNs solution of some interlayer functions as topology control and fault tolerance. On the whole, even though a certain degree of optimization can be achieved the DMA-CLD and the related OAB framework do not seem to provide high adaptivity neither from implementation nor from performance point of view. If we consider the applications mentioned in section 4 it is clear that this framework has to be further modified based on the "class" of applications addressed. For example, applications like Sustainable Bridges and Forest Fire Detection can be developed based on a subset of this framework optimized for environmental monitoring while applications like CarTalk 2000 and Sense-R-U might result in unforeseen complications and problems due to the more intricate and generic information interaction involved.

A different way of separating a "common entity" from the traditional protocol stack is presented in the idea of the Horizontal framework. In this case the separation is based on functions not on data structures. The Horizontal framework provides a separation of the functions currently covered by the different layers of the OSI model by selecting some that are not definitely related to a fixed layer and creating a new “horizontal” or “cross-layer” entity called CLM entity. This new entity has a modular structure in itself where modules are roughly corresponding to different tasks that might be related directly to network operation (topology management, energy efficient routing etc.) or might be more general and related to the single node (duty cycle determination, switching between different power modes at the node level etc.). The Data Link Layer and the Physical Layer are preserved but some of their general purpose functions are transferred to modules in the CLM entity. As a result of this organization the Horizontal Framework provides a simplification of the application/protocol stack and makes programming easier. It provides a high degree of adaptivity in a simplified structure and allows for different approaches to dealing with power efficiency issues both at the node and network level. Fault tolerance is not directly resolved. A major advantage is that it tries to balance the advantages of CL and traditional design by preserving partially the layered architecture. However, from implementation point of view the interoperability between the modules in the CLM is under question especially if their number is increased (the authors illustrate their idea with two modules). Further more the boundary between which operations or issues should be separated from the Physical and Data link and included as modules in the CLM and those which should be kept is not clearly defined. This also leads to implementation problems. However we believe that a further elaboration in this direction is very promising and might lead to resolving in an optimized way both the performance and the implementation issues. We can support this idea by using the Horizontal Framework as a generic development platform for the applications discussed. As the Sustainable Bridges and Forest Fire Detection have similar optimization parameters including similar modules in the CLM to realize these functions will provide the required adaptivity. On the other hand the addition of cross-layer module handling mobility issues can easily take into account the additional application requirements raised by adding a mobile robot in the Forest Fire Detection scenario. Furthermore, elaboration on the additional functions required by the CarTalk2000 and Sense-R-U applications can be handled partially in the application layer of the simplified stack and partially by adding new modules in the CLM. Thus it is obvious that without significant increase in the complexity new diverse application requirements can be addressed.

A very untraditional approach is presented in the XML framework. It starts from scratch and defines a totally new architecture based on the communication model and the requirements specific to WSNs. It redefines the principle of network operation based on a totally distributed approach. Each node takes a decision of participating or not participating in the network operation based on specific locally (including single node level and immediate neighborhood level) evaluated criteria. Such a conception is very straight forward and simple both from performance evaluation and implementation point of view. While it provides very high degree of adaptivity regarding different applications it does take for granted a certain high hardware standard. Nodes are aware of their location and have comparatively high computational abilities. Still this adaptivity does not come at the price of higher complexity as is the case with the other mentioned frameworks and especially TinyCubus. It resolves in an elegant way the issues of power efficiency and relation to the dynamically changing channel conditions but does not take into consideration fault tolerance. It allows for possible extensions of the selected set of parameters to include fault tolerance. Thus XLM presents a very new direction in CLD framework design which requires further research for understanding its implementation implications. Generically, the XML framework should be able to answer both the monitoring type of applications (Sustainable Bridges and Forest Fire Detection) and the more interactive ones (CarTalk 2000 and Sense-R-U). Unfortunately the authors do not provide any details on its relation to specific parameters of the application layer so it is difficult to make any remarks on that point.

5. From WSN to "smart environments"

We have so far concentrated mainly on the issues of cross-layer design related directly to WSNs. However, the future "smart environments" do not only collect information from the environment. As the definition was given in the introduction of this chapter they will "acquire and apply knowledge about the environment to improve the users' experience". Thus not only sensing nodes will be required but also "acting" nodes, known as "actuators" or "actors". While the sensor nodes are very low-power, low-cost sensing devices with very limited communication and processing capabilities the actor nodes are more resource rich nodes, equipped with better communication abilities (more processing power, larger transmission range) and longer battery life. These networks as defined in (Akyildiz & Kasimoglu, 2004) are known as Wireless sensor and actuator networks -WSAN (Fig. 7). Furthermore, while there might be hundreds or thousands of sensor nodes, very densely deployed in a given area, such a dense deployment is not expected for actor nodes. The authors discuss single actor and multi actor networks where the number of actuating devices will be strongly dependent on the specific application and the environment conditions.

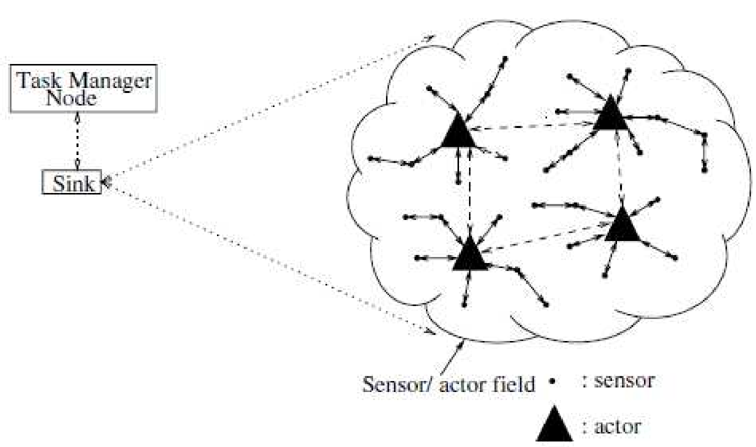


Fig. 7. The physical architecture of WSANs (Akyildiz & Kasimoglu, 2004).

WSAN have two unique features, which clearly differentiate them from WSNs: real time requirement and coordination. The real time requirement comes from the fact that WSAN are expected to immediately respond to a certain event i.e. in case of forest fire actions should be initiated immediately in order to reduce scale of damage. The coordination requirement has two aspects: one provides transmission of the event features from the

sensors to the actor nodes while the other is related to the coordination among the actor nodes themselves and the optimization of their actions.
In the survey the authors present a very detailed analysis of the specifics, requirements and open research issues related to WSAN. Together with the structure and functionalities of the future WSAN networks the authors discuss the questions of protocol design for these networks and its relation to cross-layer design. Akyildiz et al. argue that the presence of actor nodes makes protocol design even more complicated as additional operational issues like efficient communication between sensors and actors and effective coordination between actors in a multi actor network make the restrictions stricter and even protocols suitable for WSNs might be rendered insufficient They suggest a new protocol model for WSAN that is three dimensional and inherently cross-layered (Fig. 8).

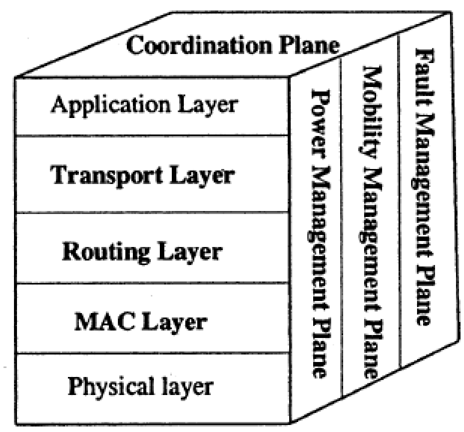


Fig. 8. WSAN protocols stack (Akyildiz & Kasimoglu, 2004).

The suggested model consists of three planes: communication plane, management plane and coordination plane. The communication plane is responsible for realizing the communication between the nodes. The data received by a node at the communication plane is submitted to the coordination plane to decide how the node should react to this data. The management plane in turn is responsible for monitoring the operation of the network and controlling the sensor and actor nodes. Important issues as mobility management, power management and fault tolerance are handled by the management plane. The coordination plane is more related to the actor nodes as they have to collaborate very efficiently with each other in order to perform a certain task, working sequentially or concurrently. It is stated that the realization of WSANs will need to satisfy more severe constraints and specific requirements introduced by the coexistence of sensor and actor nodes. A major research issue is the definition of a framework to characterize the protocol design and the suggested planes. The authors also stress on the fact that the cross-layer approach is the way to provide effective sensing, data transmission and acting.

6. Conclusion

In this chapter we have tried to discuss and summarize different issues related to cross-layer design, the new unconventional protocol design approach that has been suggested to meet the challenges and restrictions posed by the newly emerging networks like WSN and WSAN. These networks are based on small but intelligent devices (smart sensor nodes) that can sense the environment, collect data and transfer data, if necessary react to a specific event. Furthermore the operation of the network is realized as a result of the collaborative action of large numbers (few tens to thousands) of nodes. Such networks behave quite differently from the traditional IP networks: first because of the inherently unstable and unpredictable nature of the wireless channel through which the multi-hop communication is realized, second due to the great limitations of the nodes in both capacity and power and third, due to the fact that they are highly application-centric and rely on the collaborative operational model to realize a specific task. Thus, unlike conventional networks they have their own design and resource constraints. Resource constrains include the limited amount of energy available to the nodes the short communication range, the low bandwidth and very limited storage and processing. Design constraints are based on the application and may vary as the applications themselves vary from environment monitoring to health care and event detection and tracking. Furthermore, WSAN introduce questions of coordination between actors and sensors.

Numerous studies have proved that the traditional layered protocol design approach (the OSI model) is not suitable to meet these constraints and specifics. Many researchers argue that a new holistic approach is required. In this line a number of cross-layer solutions, that allow interaction between protocols at different layers have been suggested and proved to be more suitable to the protocol design for WSNs. Benefiting from the interaction between different layer higher efficiency and prolonged network lifetime can be achieved. However the advocates of cross-layer design argue that such approaches are very dangerous as they damage the modularity of the design and can result in a number of unforeseen and unwanted effects.

In this chapter we have discussed the definition of cross-layer design approach, the suggested methods and classifications in the existing literature involving cross-layer interactions as well as the problems and challenges involved. Furthermore we have explained the necessity for creating a conceptual structure for protocol design that will suit the requirements and restrictions of WSNs. A review of the few suggested so far CLD frameworks, including the TinyCubus, DMA-CLD, OAB and XLM frameworks was given. By defining criteria for their evaluation we have contrasted and compared these suggestions. The chapter was concluded with a look towards the future: from wireless sensor networks and cross-layer design issues to the “smart environments” realized by wireless sensor and actor networks.

Finally we hope that this work will throw additional light on issues related to the cross-layer design and CLD frameworks and provide a background for a future unified approach to protocol design in WSN and WSAN that researchers may want to address as they move forward.

7. References

Akyildiz, I. F.; Su, W.; Sankarasubramaniam. Y. & Cayirci, E. (2002). A Survey on Sensor Networks. *IEEE Communications Magazine,* Vol. 40, No. 8, (August 2002), (102-116), ISSN: 0163-6804

Akyildiz, I. F. & Kasimoglu, I. (2004). Wireless sensor and actor networks: research challenges. *Ad Hoc Networks,* Vol. 2, No. 4, (October 2004), (351-367), ISSN: 1570-8705

Akyildiz, I. F.; Vuran, M. C. & Akan, O. B. (2006). A Cross-Layer Protocol for Wireless Sensor Networks. *Proceedings of Conference on Information Sciences and Systems,* pp. 1102 - 1107, ISBN 1-4244-0349-9, Princeton, NJ, March 2006, Information Sciences and Systems (CISS), Princeton

Alonso, L.; Ferrus, R. & Agusti, R. (2003). MAC-PHY enhancement for 802.11b WLAN systems via cross-layering. *Proceedings of IEEE VTC-Fall,* pp. 776 - 780, ISSN : 1090-3038, Orlando, FL, October 2003

Carpenter, B. (1996). Architectural Principles of the Internet, RFC 1958, June 1996. [Online]. Available: http:/www.rfc-editor.org/rfc/rfc1958.txt

CRUISE, (2007). European IST project CRUISE, Deliverable no.:D112.1, *Report on WSN applications, their requirements, application-specific WSN issues and evaluation metrics*, IST-027738/ CRUISE

Cui, S.; Madan, R.; Goldsmith, A. & Lall, S. (2005). Joint routing, MAC, and link layer optimization in sensor networks with energy constraints, *Proceedings of IEEE ICC 2005,* pp. 725 - 729, ISBN 0-7803-8938-7, May 2005

Fang, Y. & McDonald, A. B. (2004). Dynamic codeword routing (DCR): a cross-layer approach for performance enhancement of general multi-hop wireless routing, *Proceedings of IEEE SECON 2004*, pp. 255 - 263, ISBN 0-7803-8796-1, October 2004

Hakala, I. & Tikkakoski, M. (2006). From vertical to horizontal architecture - a cross-layer implementation in a sensor network node. *Proceedings of First International Conference on Integrated Internet Ad hoc and Sensor Networks (InterSense),* Article No:6, ISBN:1-59593-427-8, Nice, France, May 2006

Kawadia, V., & Kumar, P. R. (2005). A cautionary perspective on cross-layer design. *IEEE Wireless Communications Magazine*, Vol. 12, No. 1, (February 2005), (3-11), ISSN: 1536-1284.

Lachenmann, A., Marron, P.J., Minder, D., & Rothermel, K., (2005). An Analysis of Cross-Layer Interactions in Sensor Network Applications, *Proceedings of the Second International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 121-126, ISBN: 0-7803-9399-6, December 2005.

Lee, L. T. (2006). Cross-layer design and optimization for wireless sensor Networks. *MSc Thesis, Naval Postgraduate School,* (March 2006), Monterey California

Liu, Q.; Zhou, S. & Giannakis, G. B. (2004). Cross-layer combining of adaptive modulation and coding with truncated ARQ over wireless links. *IEEE Transactions on Wireless Communications,* Vol. 3, No. 5, (September 2004), (1746 - 1755), ISSN: 1536-1276

Marrón, P. J.; Lachenmann, A.; Minder, D.; Hähner, J.; Sauter, R. & Rothermel, K. (2005a). TinyCubus: A Flexible and Adaptive Framework for Sensor Networks, *Proceedings of the 2nd European Workshop on Wireless Sensor Networks*, pp. 278-289, ISBN 0-7803-8801-1, February 2005

Marrón, P. J.; Minder, D.; Lachenmann, A. & Rothermel, K. (2005b). TinyCubus: A Flexible and Adaptive Cross-Layer Framework for Sensor Networks. *4. GI/ITG KuVS Fachgespräch "Drahtlose Sensornetze", Technical Report TR 481, Computer Science Department,,* (March 2005), (49 - 54), Zurich, Switzerland

Marrón, P. J.; Saukh, O.; Krüger, M.; & Große, C. (2005c). Sensor Network Issues in the Sustainable Bridges Project, *Proceedings of the European Projects Session of the Second European Workshop on Wireless Sensor Networks (EWSN 2005)*, Istanbul, Turkey, January 2005

Marrón, P. J.; Minder, D.; Lachenmann, A. & Rothermel, K. (2005d). TinyCubus: An Adaptive Cross-Layer Framework for Sensor Networks. *it - Information Technology,* Vol. 47, No. 2, (2005), (87 - 97), ISSN: 18125638

Melodia, T.; Vuran, M. C. & Pompili, D. (2006). The State-of-the-art in Cross-layer Design for Wireless Sensor Networks, *Lecture Notes in Computer Science (LNCS)*, Vol. 3883/2006, (May 2006), (78–92), ISSN 0302-9743

Morsink, P.; Hallouzi, R.; Dagli, I.; Cseh, C.; Schafers, L.; Nelisse, M.; & Bruin, D. D. (2003). Cartalk 2000: Development of a cooperative adas based on vehicle-to-vehicle communication, *Proceedings of the 10th World Congress on Intelligent Transport Systems and Services (ITS)*, Madrid, Spain, November 2003.

Raisinghani, V. T. & Iyer, S. (2004). Cross-layer design optimization in wireless protocol stacks, *Computer Communication*, Vol. 27, No.8, (May 2004), (720—724), ISSN: 0140-3664.

Safwat, A. M. (2004). A novel framework for cross-layer design in wireless ad-hoc and sensor networks, *Proceedings of IEEE GlobeCom Workshops*, pp. 130-135, ISBN: 0-7803-8798-8, December 2005

Shakkottai, S.; Rappaport, T. S. & Karlsson, P. C. (2003). Cross-layer design for wireless networks. *IEEE Communications Magazine,* Vol. 41, No. 10, (October 2003), (74 - 80), ISSN: 0163-6804

Sichitiu, M. L. (2004). Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks, *Proceedings of IEEE INFOCOM 2004*, pp. 1740 - 1750, ISSN: 0743-166X, March 2004

Sokullu, R., & Karaca, L. O., (2009). Simple and Efficient Cross-Layer Framework Concept for Wireless Sensor Networks, *Proceedings of 12th International Symposium on Wireless Personal Multimedia Communications*, pp.40, Sendai, Japan, September 2009.

Srivastava, V. & Motani, M. (2005). Cross-layer design: A survey and the road ahead. *IEEE Communications Magazine,* Vol. 43, No. 12, (December 2005), (112 - 119), ISSN: 0163-6804

Stallings, W. (2006). *Data and Computer Communications,* Prentice Hall, ISBN 13: 9780132433105

Tian, J. & Coletti, L. (2003). Routing approach in CarTALK 2000 project, *Proceedings of 12th IST Mobile and Wireless Communications Summit 2003*, Paper No. 1047, Aveiro, Portugal, June 2003

van Hoesel, L.; Nieberg, T.; Wu, J. & Havinga, J. M. (2004). Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *IEEE Wireless Communications,* Vol. 11, No. 6, (December 2004), (78-86), ISSN: 1536-1284

Vuran, M. C.; Gungor, V. B. & Akan, O. B. (2005). On the interdependency of congestion and contention in wireless sensor networks, *Proceedings of SENMETRICS '05*, pp.136 – 147, San Diego, CA, July 2005

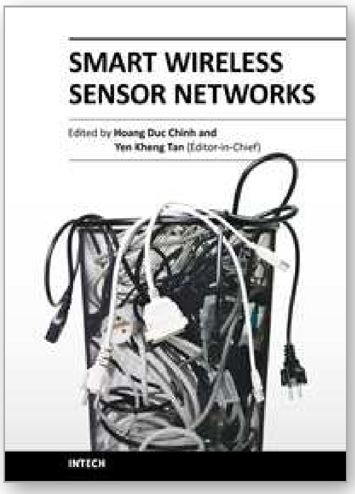
Wang, Q., & Abu-Rgheff, M. A. (2003). Cross-Layer Signalling for Next-Generation Wireless Systems. *Proceedings of IEEE Wireless Communications and Networking Conference 2003 (IEEE WCNC 2003*, pp. 1084-1089, ISSN : 1525-3511, New Orleans, LA, USA, March 2003

Yick, J., Mukherjee, B., & Ghosal, D., (2008). Wireless sensor network survey, *Computer Networks,* Vol.52, No.12, (August 2008), (2292-2330), ISSN: 1389-1286.

Zhang, Y., & Cheng, L., (2003). Cross-Layer Optimization for Sensor Networks, *Proceedings of 3 rd New York Metro Area Networking Workshop*, New York City, September 2003.

Zhang, Q., Zhang, Y. (2008). Cross-Layer Design for QoS Support in Multi-hop Wireless Networks. *Proceedings of IEEE 2008*, pp.64-76, ISSN : 0018-9219, January 2008.

Zhao, N. & Sun, L. (2007). Research on Cross-Layer Frameworks Design in Wireless Sensor Networks, *Proceedings of the Third International Conference on Wireless and Mobile Communications ICWMC '07*, pp. 50a - 50a, ISBN 0-7695-2796-5, Guadeloupe, March 2007.



Smart Wireless Sensor Networks
Edited by Yen Kheng Tan

ISBN 978-953-307-261-6
Hard cover, 418 pages
Publisher InTech
Published online 14, December, 2010
Published in print edition December, 2010

The recent development of communication and sensor technology results in the growth of a new attractive and challenging area â€“ wireless sensor networks (WSNs). A wireless sensor network which consists of a large number of sensor nodes is deployed in environmental fields to serve various applications. Facilitated with the ability of wireless communication and intelligent computation, these nodes become smart sensors which do not only perceive ambient physical parameters but also be able to process information, cooperate with each other and self-organize into the network. These new features assist the sensor nodes as well as the network to operate more efficiently in terms of both data acquisition and energy consumption. Special purposes of the applications require design and operation of WSNs different from conventional networks such as the internet. The network design must take into account of the objectives of specific applications. The nature of deployed environment must be considered. The limited of sensor nodesâ€™ resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design. A smart wireless sensor network must be able to deal with these constraints as well as to guarantee the connectivity, coverage, reliability and security of networkâ€™s operation for a maximized lifetime. This book discusses various aspects of designing such smart wireless sensor networks. Main topics includes: design methodologies, network protocols and algorithms, quality of service management, coverage optimization, time synchronization and security techniques for sensor networks.

Fax: +385 (51) 686 166
www.intechopen.com

Fax: +86-21-62489821