

Cross-layer Optimisation for Quality of Service Support in Wireless Sensor Networks

Mahsa Mohaghegh
Massey University/SEAT,
Auckland, New Zealand;
Email: M.Mohaghegh@massey.ac.nz

Chris Manford
Department of Computing, Unitec,
Auckland, New Zealand;
Email: Cmanford@unitec.ac.nz

Abdolhossein Sarrafzadeh
Department of Computing, Unitec,
Auckland, New Zealand;
Email: hsarrafzadeh@unitec.ac.nz

Abstract—Wireless sensor networks need to deliver real-time services such as video, audio and traditional data services therefore providing efficient quality of services (QoS) support is essential. In this paper we aim to address the time-delay parameter of QoS this is implemented using a new cross-layer framework design. The concept of cross-layer design is based on architecture where different layers can exchange information in order to improve the overall network performance. Promising results achieved by cross-layer optimization initiated significant research activity in this area.

We present results from simulations of the new cross layer design and traditional OSI model using the OMNET++ software. We show that the cross layer design provides a feasible and flexible approach to solving the conflict between different layers in a standard OSI model. We demonstrate that the cross layer optimization is a promising solution and that enhances the quality of service in wireless sensor network applications.

Keywords—Cross-layer design, Wireless Sensor Networks, Quality of Service

I. INTRODUCTION

Recent advantages in wireless communication and electronics have caused an increase in the development of large scale sensor networks and research into different possible usages. A growing number of WSN applications demand real-time quality of service (QoS) guarantees. Such QoS requirements usually include survivability, resilience and scalability. The main challenge of satisfying these requirements is to minimize network energy consumption, so as to maximize sensor network lifetime.

Notwithstanding, the increased interest and growth of WSN some specific limitations still remain. These include the limited bandwidth available, low dry cell capacity and limited processor and memory capacity. These limitations can cause some deficiency in quality of service support.

The success of layered architecture has been observed in many fields. However, the open system interconnect (OSI) architecture of computer networks may not be able to provide QoS in all parameters, so optimization goals need to pursue in multiple layers. Also, since some applications do not need all layers, and optimizations may conflict with each other, the general notation “Cross-Layer Design” suggests the breaking

of OSI hierarchical layers in communication networks. In this paper we study the application of cross-layer protocol design and optimization. To achieve this we clarify and justify the optimization goals targeted in this research, and justify our motivation to take the cross-layer optimization approach to achieve these goals.

The aim of the study is to provide a feasible approach to enhance the QoS in multi-purpose wireless sensor networks. Accordingly, the contributions of this paper are as follows:

Firstly, we explain cross-layer as a coordinator of quality-of service mechanism between several different layers.

Secondly, we explain cross-layer usage and implementation by OMNET++ [1].

Next, we show experiments and model evaluation, followed by implementation of cross-layer design in each node using OMNET++ software.

Finally, the results of our experiments with cross-models and OSI-models in both uniform and non-uniform conditions will then be evaluated, compared and discussed, and our conclusions stated.

To the best of our knowledge, this is the first work that provides a cross-layer design of end-to-end package re-ordering in wireless sensor networks.

A. Related Work

While a significant amount of research has been done on several important aspects of WSNs (such as architecture and protocol design, energy conservation, and locating), supporting Quality of Service (QoS) in WSNs is still a largely unexplored research field. This is mainly because WSNs are very different from traditional networks. The term *QoS* is used too frequently, and can mean and relate to a number of different things[2].

QoS challenges from general wireless networks can be listed as follows: 1.) Elimination of network limitations to achieve network efficiency in hardware layer energy consumption; 2.) Improve system performance such as possible theoretical analysis solutions, together with discussion of scalability solutions; 3.) Achieving necessary requirements for quality of service; 4.) Improved utilization of resources.

Recently, cross-layer network design has been investigated in order to achieve QoS parameters in different aspects[3].

Cross-layer design aims to identify the relationship between network parameters and the distribution of end-to-end delay,

and accordingly influence the design of real-time solutions for WSNs.

B. Goals and Motivation

The benefits of a cross-layer design aim at coupling the functionality of network layers and as a consequence boosting the system performance. One efficiency idea that has recently been offered in the area of quality of service support in wireless sensor networks is the incorporation of cross-layer design in these networks [4, 5]. With the help of a cross-layer framework, we can enhance QoS, especially in the area of resource constraints.

Three main problems must be considered before implementing a cross-layer framework within a WSN: The first problem, or optimization target, is how to maintain the efficiency of each node, and how to support a large scale network while keeping the optimum situation[6]. The second problem is how to enable optimum power efficiency within the sensor. The third problem is how to optimize multi-tasking within the nodes when their sources are limited and often single-purpose[7]

II. CROSS-LAYER AS A COORDINATOR MECHANISM FOR QUALITY OF SERVICE IN SEVERAL LAYERS

We can use cross-layer to coordinate between two or more mechanisms which act in several layers[8]. However, it is possible that these mechanisms, when used in conjunction with each other, can reduce the quality service for the total network. We modeled two FIFO simple-priority queues in both the application and the data layer, which were used to optimize the quality of service based on priority. The problem is that priorities are not passed between layers so packets with high priority in the application layer are considered to have normal priority in the data-link layer and will be queued and processed the same as normal packets. This causes processing delay [9]. Using the cross-layer design suggested we will be able to coordinate priorities in two queues and remove unwanted receive and transmit delays in the data-link layer queue.

III. EXPERIMENTS AND MODEL EVALUATION

In this section we report on the design of experiments that evaluate the performance of a cross model in comparison with the OSI model. The main emphasis of our experiments is on the delay of packets from each node to a sink node, which is one of the QoS parameters. After designing some experiments on a single node, we construct experiments on a sensor network containing several nodes and a sink node.

A. Cross-layer Implementation in Node by OMNET++ Software

Consider a sensor network architecture in which a simple priority transmit queue in the application layer has its own priority and a transmit queue in the data-link layer with another priority [10]. The function of these queues is to arrange high-priority packets to the front of the queue and send them faster than normal-priority packets. Packets that contain node sensor behavior commands, node or link damage reports and energy shortage reports in several nodes will have

first priority in the data-link layer. Other packets that contain sensor information from the operational environment will have second priority. In the application layer, packets with node situational information commands have first priority, while other packets have second priority. In each layer, priority is given to packets that are from a higher class and have higher priorities.

The greatest problem in this model is caused by the fact that a packet which has a higher priority in the application layer is considered a normal-priority packet in the data link layer. In this situation, a high-priority application layer data packet which should be processed faster or moved to a neighbor node may be queued in the data-link layer behind normal-priority packets and delayed.

The main purpose of applying cross-layer is to optimize this model and allow communication to and from the application and data-link queues between multiple layers [5].

The total transmission delay for each layer for the application and data link layers is calculated by equation (1). The total transition delay by cross-layer is shown by equation (2)

$$T_u = \left(\sum_{i=1}^{MaxQ} P_{fi} \times i + \sum_{i=1}^{MaxQ} P_{li} \times i \right) \times T_{tr} \quad (1)$$

$$T_0 = \left(\sum_{i=1}^{MaxQ} P_{fi} \times i \right) \times T_{tr} + T_{cross} \quad (2)$$

$MaxQ$ stands for maximum number of packets that can be stored in the transmission priority queue.

P_{fi} stands for first class packet's probability in the application layer that are located after (i) high-priority packets of data link layer.

P_{li} stands for first class packets in the application layer, located after (i) normal-priority packets from data link layer that are inserted into the data link layer queue.

T_u stands for default average delay for first class packets from the application layer in the data link layer.

T_{tr} stands for average transmission delay for each packet from data link layer.

T_{cross} stands for average time to interact each packet with cross-layer and T_0 is the optimized delay for this packet.

Generally, $MaxC$ stands for priority levels in the situation with more than two priority levels.

P_{cji} stands for probability of insertion of a C class packet from the application layer after the i-j classes in the data link layer.

Relations (3), (4) show how we can estimate the average time for both the Cross and OSI models.

$$T_u = \left(\sum_{j=1}^c \sum_{i=1}^{MaxQ} P_{cji} \times i + \sum_{j=c+1}^{MaxC} \sum_{i=1}^{MaxQ} P_{cji} \times i \right) \times T_{tr} \quad (3)$$

$$T_0 = \left(\sum_{j=1}^c \sum_{i=1}^{MaxQ} P_{cji} \times i \right) \times T_{tr} + T_{cross} \quad (4)$$

Packets are divided into four classes – A, B, C, and D. These show the priority of each packet in the data-link layer and the application layer. Each class is described as follows:

Class A: packets with first priority in both data-link and application layer.

Class B: critical packets with first priority in application layer and second priority in data-link layer.

Class C: packets with second priority in application layer and first priority in data-link layer.

Class D: packets with second priority in both the application and data-link layer.

In our experiments each packet contains some parts which are used for different purposes. One of these parts is the *Index* of the packet, which determines the priority of this packet in application and data-link. The other part of a packet is the time of generation which is used by the sink node to calculate the time of receiving the packets.. The main part of a packet is the data which is transferred to the sink node. Fig. 1 shows the structure of the packets in our experiments.

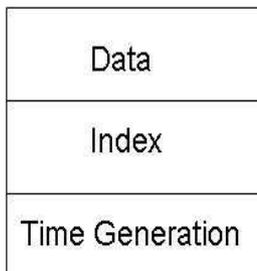


Fig. 1 Structure of packets

In the first step of packet transmission we arrange packets based on their priority in the application layer. Firstly, packets of class B are placed in the front of the queue and after them packets of classes A, C and D.

Secondly, the sorted packets are transmitted to the data-link layer and they are rearranged based on their priority in the data-link layer.

In the OSI model, packets of class B are placed after packets of classes A and C. Packets of class B which were at the front of the queue in the application layer are now placed in the middle of the queue in the data-link layer. In this way, transmission of class B packets (which have high priority in application layer) is delayed. This is contradictory with QoS parameters. Figure (2) shows packet transmission based on the OSI model.

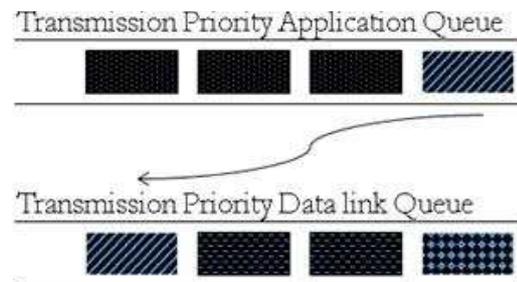
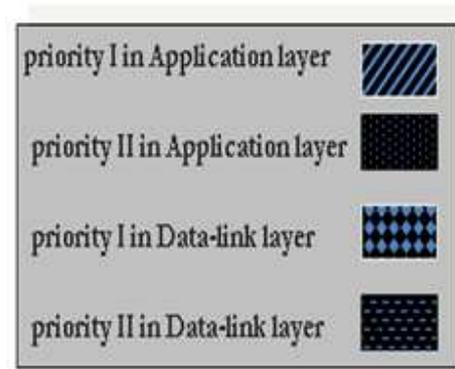


Fig. 2 Packet transmitting based on OSI model

In the Cross model, in step 2, critical packets (Class B) are not placed in the middle of the queue. In this way, the priority is now assigned to critical packets in the data-link layer and transmission of class B packets is not delayed. Figure (3) shows packet transmitting based on the Cross model.



Fig. 3 Packet transmitting based on Cross model

1. Comparison of proposed Cross model and OSI model in Uniform Condition

In the first experiment, we consider four classes (A, B, C and D) which have equal probability values, so we defined this model's condition as *uniform*. Figure (4) shows the average time to process class B packet transmissions in both the OSI and Cross models in uniform conditions. In this figure the horizontal axis shows the number of packets in queue that each node can transmit. In this experiment the range of this axis is 5 to 50. As shown, the Cross model outperforms the OSI model in all cases.

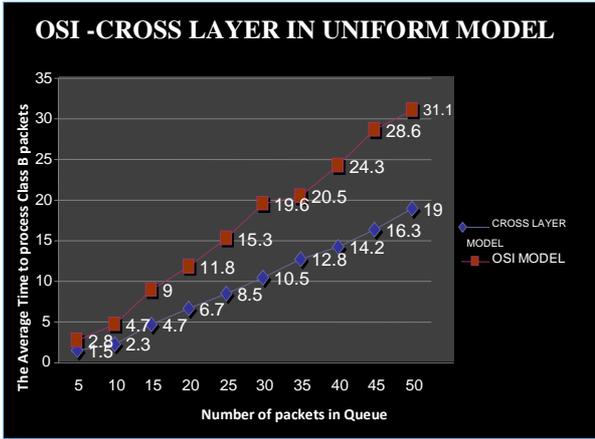


Fig. 4a Average time of packet transmission (Class B)

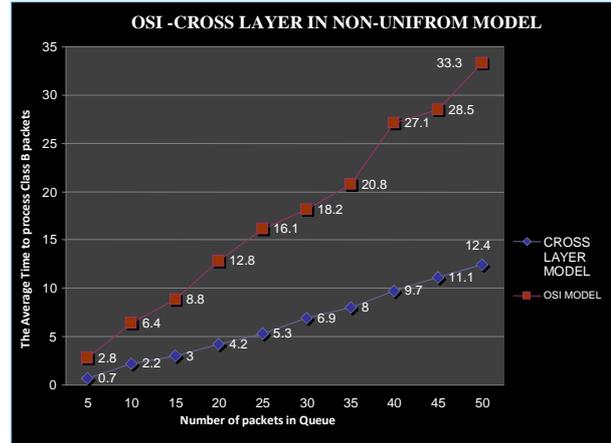


Fig. 5a Performance of Cross model outperforms OSI model in non-uniform condition with large number of packets

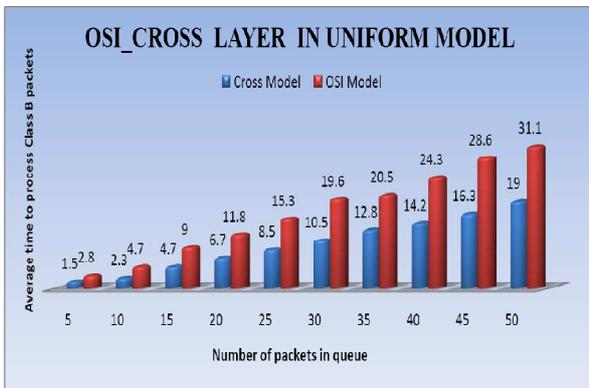


Fig. 4b Average time of packet transmission (Class B)

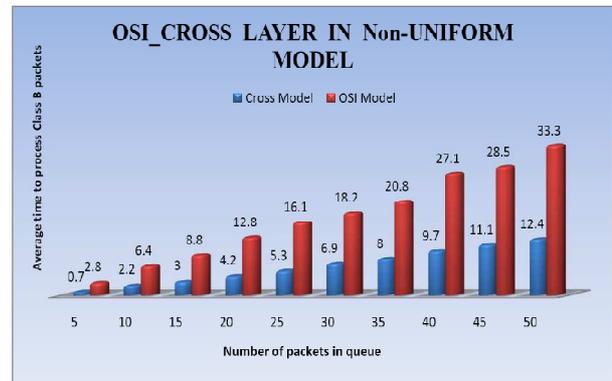


Fig. 5b Performance of Cross model outperforms OSI model in non-uniform condition with large number of packets

2. Comparison of proposed Cross model and OSI model in Non-Uniform Condition

In the second experiment we consider four classes (A, B, C and D) which have different probability values, so we define this model's condition as *non-uniform*. In this experiment we assign the following values to the four classes. $P(A)=10\%$, $P(B)=30\%$, $P(C)=40\%$, $P(D)=20\%$

These assigned values can change in different experiments. If the probability of critical packets (Class B) increases, then the difference between the performance of applying the Cross model and the OSI model increases too.

Fig. 5a and 5b show the average time to process a class B packet transmission in the non-uniform condition. As shown, the Cross model outperforms the OSI model in different all packet conditions.

Fig. 5a and 5b also both show that the average time of transmission in both the Cross and OSI models increases with an increasing number of packets in each node. But the gradient of the slope shown in the OSI model is much steeper than the slope shown in the Cross model (Fig. 5a). It shows that the performance of the Cross model exceeds the OSI model in the non-uniform condition with a large number of packets.

The results of the two previous experiments show that based on the average time of sending data packets, the Cross model processes packets more effectively than the OSI model. This difference becomes more prominent in non-uniform conditions.

Most problems in a real-world situation follow the non-uniform model, so we can expect that the Cross model achieves better performance in these types of problems.

An important fact to note about this simulation is that all parts of it are implemented in OMNET++ software [1] and we obtained the results based on 30 different runs in OMNET environment.

B. Implementation of a Sensor Network by Applying Cross and OSI Models

In contrast with the previous experiments, where most of the emphasis was on the average time of sending data packets by each sensor node, this experiment focuses more on the average time of *receiving* packets in a sensor network.

For this purpose, we first design a sensor network in OMNET++ with a fixed number of sensor nodes. The positions of these nodes are created at random in a rectangular environment. In the center of this environment we implement a sink node. A sink node's function is to receive and process data packets sent from the sensor nodes.

In each sensor node we use two parallel models separately (OSI and Cross models). In applying this strategy, we compare OSI and Cross models while using the same sequence for sending data packets. In each sensor node we generate a number of packets with non-uniform distribution to be sent to the sink node. These generated packets are sent to the sink node in two ways. One way is based on the OSI model while the other is based on the Cross model. After receiving the packets, the sink node processes them. We calculate the average time to receive critical packets (Class B) by the sink node, with results from both the OSI and Cross Models.

Figure (6) shows the average time of packet transmission in a sensor network. As can be seen, using Cross-layer is more efficient than using the OSI model. The information shown in this figure is the average of the results achieved by 30 different runs. In this experiment we assume that each node sends 10 packets to the sink node with non-uniform distribution and we use the multi-hop method for creating a connection between the nodes.

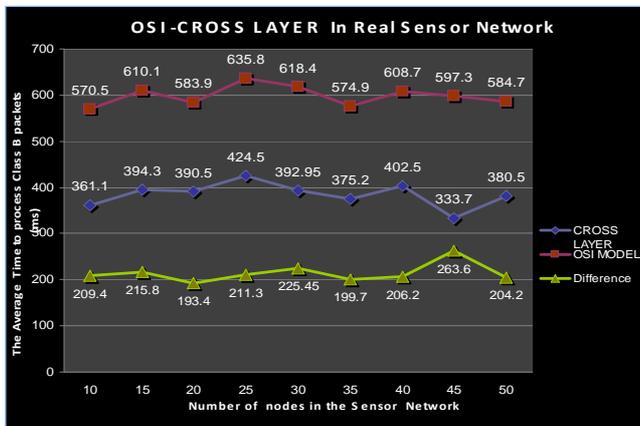


Fig. 6 Average time of packet transmission in a sensor network

In this figure we include a third curve showing the difference between the average times of packet transmission in the OSI and Cross models. In other words, this curve shows the level of performance improvement that we achieved by using the Cross model.

In the next experiment, we increase the number of packets to 30 and try to find the effect this packet increase has on the average transmission time.

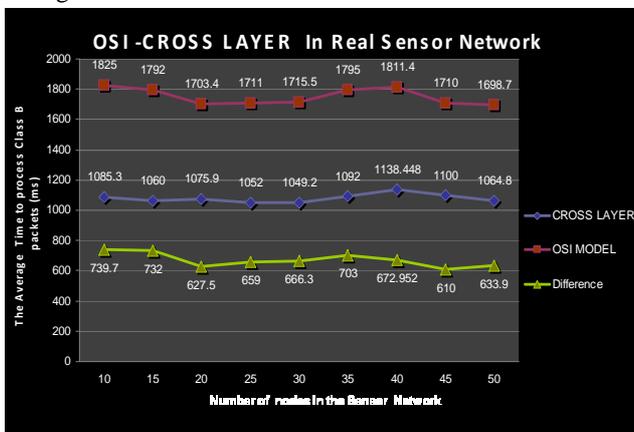


Fig. 7 Average time of packet transmission in a sensor network with 30 Packets

Both figure (6) and figure (7) show that even with an increased number of packets in a sensor network the Cross-layer method is more efficient than the OSI model.

Finally, figure (8) plots the number of nodes and the number of packets in each node against the average packet transmission time.

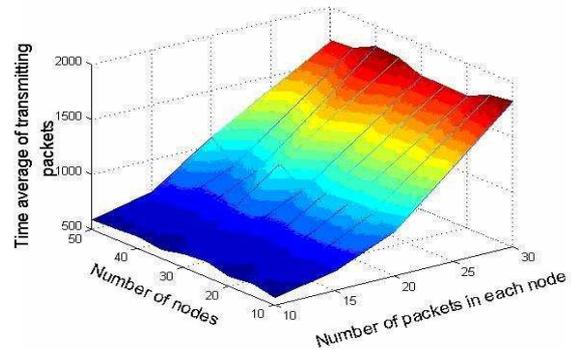


Fig. 8 Effect of number of nodes and number of packets on time average criteria

IV. CONCLUSION

In this paper, we examined the application of the Cross model in a sensor network. This was achieved by implementing a sensor network with OMNET++ software applying the OSI and Cross models separately in each node and comparing the average time of packet transmission in both cases. In all of the designed experiments the Cross model outperforms the OSI model in the time delay criteria (one of the QoS parameters).

REFERENCES

- [1] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," 2008, p. 60.
- [2] Y. Wang, M. Vuran, and S. Goddard, "Cross-Layer Analysis of the End-to-End Delay Distribution in Wireless Sensor Networks," 2009, pp. 138-147.
- [3] L. Lee and N. P. S. M. CA, "Cross-layer design and optimization for wireless sensor networks," 2006.
- [4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, pp. 2292-2330, 2008.
- [5] Y. Zhang and L. Cheng, "Cross-layer optimization for sensor networks," 2003, pp. 247-252.
- [6] Q. Zhang and Y. Q. Zhang, "Cross-layer design for QoS support in multihop wireless networks," *Proceedings of the IEEE*, vol. 96, pp. 64-76, Jan 2008.
- [7] N. Chilamkurti, S. Zeadally, A. Vasilakos, and V. Sharma, "Cross-Layer Support for Energy Efficient Routing in Wireless Sensor Networks," *Journal of Sensors*, vol. 2009, 2009.
- [8] K. Phan, R. Fan, H. Jiang, S. Vorobyov, and C. Tellambura, "Network lifetime maximization with node admission in wireless multimedia sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, 2009.
- [9] V. Kawadia and P. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Communications*, vol. 12, pp. 3-11, 2005.
- [10] C. Hager, D. Shyy, and J. Ma, "Cooperative Cross-Layer Design for Wireless Networks," *Journal of Communications*, vol. 3, p. 49, 2008.