2008 International Conference on Computer and Electrical Engineering

A Cross Model Approach to Support of QOS Parameters in Wireless Sensor Networks

Behzad Zamani
Islamic Azad University, ShahreKord Branch
bzamani@iust.ac.ir

Mehdi Mohammadi, Azarakhsh Jalalvand, Reza Fakouri, Mahmood Fathy
Iran University of Science and Technology
{mh_mohammadi,mahfathy}@iust.ac.ir,
{az_jalalvand, fakouri}@comp.iust.ac.ir

Abstract

Recent advantages in wireless communication and electronics, lead to a large scale of sensor networks development and different usages possible. Specific traits of these networks such as dynamic treatment, large scale, long life-time and being multi-purpose with specific limitations of these kinds of networks such as limited bandwidth, low dry cell capacity, limited processor and memory are caused some difficulties in quality service support in sensor networks and deficiency in cultural service quality models. Cross-layer design in networks to obtain optimum gain from sources is recently regarded. There are four categories in this technical research: First, tranquilization of limitations whereas all hardware layers are consume efficient in network energy. Second, yield system improvements are such as theoretical analyzes possible solutions and scalability solutions discussed. Third is quality requirement compliance. Fourth, earn the best sources impart.

1. Introduction

One of the most efficient ideas which are offered recently is service quality support in wireless sensor networks in cross layer design of these networks [1, 3]. This manner is shortly shown that is able to be basic or even necessary solution in networks debugging in order to support quality of service necessities in sensor networks which involved with hard source limitations.

The main target of cross layer design is to earn a practical and flexible tool to interact between necessities and limitations of new sensor networks .this target may contains three parts: the first problem is that how to keep the efficiency of each node and support a high scale networks [2]. The second problem is that how to make balance between power usage and power productivity. The third problem is multi usage optimization of nodes, when their sources are absolute and limited [6]. In section 2 cross layer is presented as a coordinator between qualities of service mechanism in several different layers. In section 3 we introduce cross layer usage and implementation by OMNET++ [7]. Section 4 contains the manner of OSI model in different test conditions and finally in section 5 the result of our arguments will be discussed.

2. Cross layer as coordinator mechanism for quality of service in several layers

We can use cross layer to coordinate between two or more mechanism, which act in several layers. [4] Because maybe are these mechanism same or against each other and reduce quality service in total networks. We imagine that two FIFO simple priority queue in application and data layer where used to optimize quality of service base on priority.

The problem is that the packets with high priority in application layer may be considered as normal packet in data- link layer and will be queued and processed like other packets. This will cause process delay [5].

By the suggested cross layer mechanism, it will be possible to coordinate priorities in two queues and deny unwished receive and transmit delays in data link layer queue.

3. Experiments and model evaluation

In this section we design some experiments which evaluate the performance of Cross model in comparison with OSI model. The most emphasis of our experiments is on the delay of packets (one of the QOS parameters) from each node to sink node. We design some experiments on a single node and after that we design some experiments on a sensor network which contains some single nodes and a sink node.

978-0-7695-3504-3/08 $25.00 © 2008 IEEE
DOI 10.1109/ICCEE.2008.58

3.1. Cross layer implementation in node by OMNET++ software

In a sensor network architecture, in order to regard priorities in transmit and receive packets, imagine a simple priority transmit queue in application layer and a transmit queue with another priority in data-link layer.

The function of these queues is to arrange high-priority packets in front of queue and send them faster than normal priority queue.

Packets which contain node sensor behaviour commands, nodes or links damage reports, energy shortage reports in several nodes have first class priority in data-link layer and other packets which contain observed information from operational environment will be second class. Packets with node situational information commands have first priority and other packets have second priority in application layer .In each layer priority is owned to packets that are from higher class and have higher priorities.

Treatable subject is this model and same situations are a packet which has higher priority in application layer is considered as a normal packet in data link layer. In this situation the packet as for application layer should be fast process or should be moved to neighbour node might be queued in data link layer after normal packets and be delayed.

The main purpose of applying cross layer is to optimize this model and coordinate one to one between application and data-link queues [3].

Total transmission delay for each layer by application and data link is calculated by Equation .1.Total transition delay is shown by Equation .2 by cross layer.

$$T_u = \left(\sum_{i=1}^{MaxQ} P_{Ii} \times i + \sum_{i=1}^{MaxQ} P_{IIi} \times i \right) \times T_{tr} \tag{1}$$

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

Max Q stands for maximum number of packets that can be storage in transmission priority queue.

P_{Ii} stands for a first class packets probability in application layer, which located after i^{th} high priority packets of data link layer. P_{IIi} stands for first class packets in application layer, which located after i^{th} normal packets from data link layer that is inserted to data link layer queue.

T_u stands for default average delay for first class from application layer in 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 optimized delay for this packet.

Generally, max C stands for priority levels for the situation with more than two priority levels.

P_{Cij} stands for probability that insert a C class packet from application layer after i to j class from data link layer.

Equations .3 and .4 show how we can estimate average time for both cross and OSI model.

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

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

Four different classes are considered for packets: Classes A, B, C, D, which represent the priority of packets in data link and application layers. Each packet is described as bellow:

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 application and data link layer.

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

In the first step of packets transmitting, we sort packets based on their priority in application layer. In this case packets of class B are in the front of the queue and after them packets of Classes A, C and D are placed.

In second step, the sorted packets are transmitted to data-link layer and they are resorted based on their priority in data-link layer.

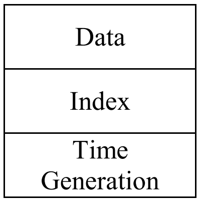

Figure 1. Structure of packets

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

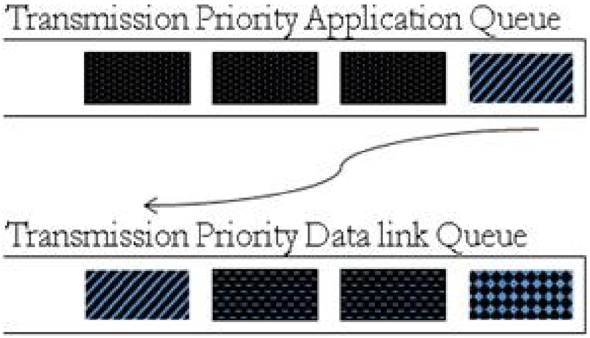

Figure 2. Packet transmitting based on OSI model

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

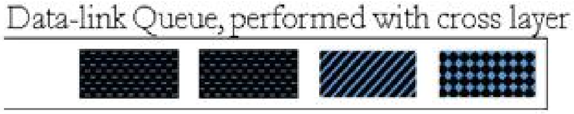

Figure 3. Packet transmitting based on Cross model

3.1.1. Comparison of proposed Cross model and OSI model in uniform condition

In first experiment, we consider, 4 classes *A, B, C* and *D* have equal probability values. So we named this method as uniform. $P_i = 25\%; i \in \{A, B, C, D\}$

Figure .3 shows time average of packet transmitting of class *B* in OSI and Cross models. In this figure, horizontal axis shows number of packet which each node can transmit. Range of this axis is 5 to 50 in this experiment. As it is shown in this figure Cross model outperforms OSI model in all cases.

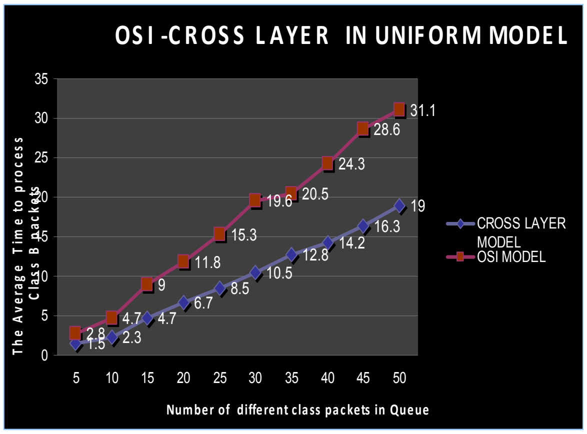

Figure 4. time average of packet transmitting (class B)

3.1.2. Comparison of proposed Cross model and OSI model in non-uniform condition

In second experiment, we consider 4 classes *A, B, C* and *D* have different probability values. So we named this model as non-uniform model. In this experiment we assign following values to four classes. *(P(A)=10%, P(B)=30%, P(C)=40%, P(D)=20%)*

These considered values can change in different experiments in the other word these values are completely optional. If the probability of critical packets (Class *B*) increases then the difference between performance of applying Cross model and OSI model, increases too.

Figure .4 shows the time average of packet transmitting of class *B* in non-uniform condition. As it is shown in this figure, cross model outperforms OSI model in different number of packet conditions.

Figure .4 shows that, the average time of transmitting in Cross and OSI models grow with increasing number of packets in each node. But the slope of growth in Cross model is very slower than slope of growth in OSI condition. It shows that the performance of Cross model outperforms OSI model in non-uniform case in large number of packets.

The results of two previous experiments show that, based on time average of packet sending criteria, Cross model is more useful than OSI model. This preference becomes more prominent in non-uniform condition.

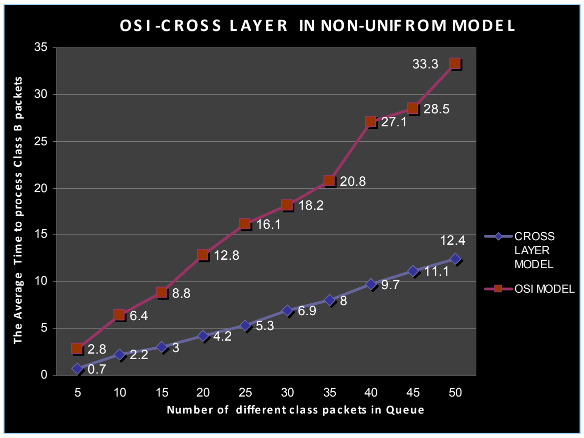

Figure 5. performance of Cross model out-performs OSI model in non-uniform case in large number of packets

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

An important tip about this simulation is that all part of this simulation is implemented in OMNET++ software [7] and we obtain mentioned results based on 30 different runs in OMNET environment.

3.2. Implementation of a sensor network with applying Cross and OSI models

In contrast with previous experiments which most emphasis was on average time of sending data in each sensor node, in this experiment the most emphasis is on the average time of receiving packets in a sensor network.

For this purpose, at first we design a sensor network in OMNET++ with a fix number of sensor nodes. The position of these nodes is created at random in a rectangular environment. In the center of this environment we place sink node. Receiving and processing of packets is the task of sink node in a sensor network.

In each node we use two parallel models separately (OSI and Cross models). With applying this strategy, we compare OSI and Cross models in a same packet sequence. After that in each we generate some packets with non-uniform distribution and these generated packets are sent to sink node in two ways. One way is based on OSI model and the other one is based on OSI model and in the sink node the received packets is processed. Now we calculate the average time of receiving critical packets (Class B) in sink node based on OSI and Cross Models.

Figure .5 shows the time average of transmitting packets in a sensor network. As it can be seen, in this figure using cross layer is more efficient than using OSI model. The results of this figure achieve by average of 30 different runs. In this experiment we consider that each node sends 10 packets to sink node with non-uniform distribution and we use multi hope method for creating connection between nodes.

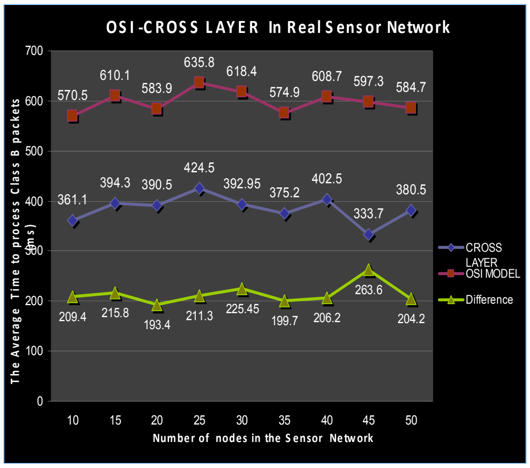

Figure 6. Time average of transmitting packets in a sensor network

In this figure we proposed third curve which is the difference between average time of transmitting packets in OSI and Cross models. In the other word this curve shows the amount of performance that we achieved by using Cross model.

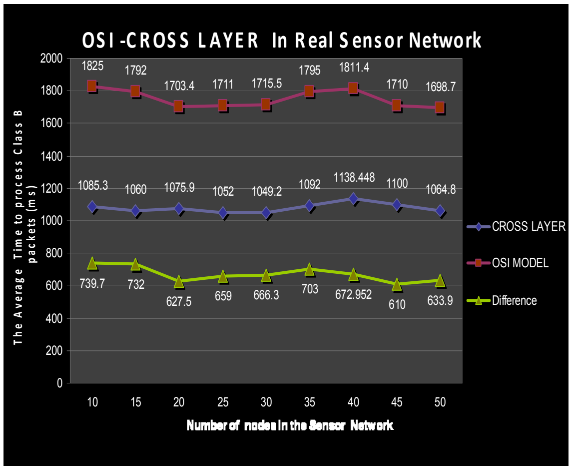

Figure 7. Time average of transmitting packets in a sensor network

In next experiment, we increase number of packets in each node (30 packets in each node) and try to find the effect of number of packet in each node on time average criteria.

Figure .5 and Figure .6 show that with increasing number of packets in a sensor network, Cross layer is more efficient than OSI model.

Finally, Figure .7 presents effects of the number of nodes and the number of packets in each node on time average of packet transmitting criteria. As it is shown in this figure number of packets in each node is more effective than number of nodes in a sensor network. Figure .8 represents the effect of number of nodes and number of packets on time average criteria.

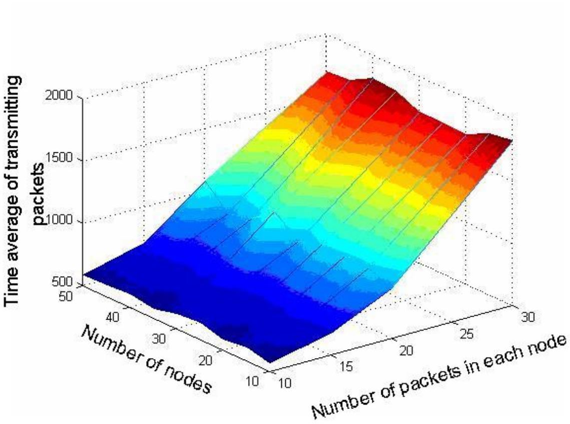

Figure 8. Effect of number of nodes and number of packets on time average criteria

4. Conclusions

In this paper, we examined applying Cross model in a sensor network. For this purpose, we implement a sensor network with OMNET++ software and apply OSI and Cross models in each node separately. Then compare average time of transmitting packets in these two cases. In all of the designed experiments, Cross model outperforms OSI model in time delay criteria (one of the QOS parameters)

5. References

[1] I.F. Akyildize et al., "Wireless Sensor Networks: a survey, "*Computer Networks*, Vol. 38, pp. 393-422, March 2002.

[2] H. Karl and A. Wilig, "A short survey of wireless sensor networks, "*Technical Report TKN-03-018, Tele-communication Networks Group*, Technical University Berlin, October 2003.

[3] Y. Zhang and L. Cheng, "Cross-Layer optimization for sensor networks", *New York Metro Area Networking Workshop 2003*, New York City, NY, USA, September 12, 2003.

[4] Hoojoong Kwon, Tae Hyun Kim, Sunghyun Choi, and Byeong Gi Lee, "Cross-Layer Lifetime Maximization under Reliability and Stability Constraints in Wireless sensor Networks", *in Proc. IEEE ICC 2005*, Seoul, Korea, May 16-20, 2005

[5] V. Kawadia and P. R. Kumar, “A cautionary perspective on cross layerdesign,” *IEEE Wireless Commun*. Mag., vol. 12, no. 1, pp. 3–11, Feb.2005.

[6] L. Song and D. Hatzinakos, “Energy efficiency limits of broadcasting inwireless networks,” *IEEE Trans.Wireless Communications*,May 2005,submitted for publication

[7] OMNeT++ website, http://whale.hit.bme.hu/omnetpp/.