Copyright 2012 KSII

Cross-Layer Architecture for QoS Provisioning in Wireless Multimedia Sensor Networks

Muhammad Omer Farooq[1], Marc St-Hilaire[2] and Thomas Kunz[2]
[1]Institute of Telematics
University of Lübeck, Germany
[2]Department of Systems and Computer Engineering
Carleton University, Ontario, Canada
e-mail: farooq@itm.uni-luebeck.de, {marc_st_hilaire, tkunz}@sce.carleton.ca

Received August 5, 2011; revised December 9, 2011; accepted December 26, 2011; published January 31, 2012

Abstract

In this paper, we first survey cross-layer architectures for Wireless Sensor Networks (WSNs) and Wireless Multimedia Sensor Networks (WMSNs). Afterwards, we propose a novel cross-layer architecture for QoS provisioning in clustered and multi-hop based WMSNs. The proposed architecture provides support for multiple network-based applications on a single sensor node. For supporting multiple applications on a single node, an area in memory is reserved where each application can store its network protocols settings. Furthermore, the proposed cross-layer architecture supports heterogeneous flows by classifying WMSN traffic into six traffic classes. The architecture incorporates a service differentiation module for QoS provisioning in WMSNs. The service differentiation module defines the forwarding behavior corresponding to each traffic class. The forwarding behavior is primarily determined by the priority of the traffic class, moreover the service differentiation module allocates bandwidth to each traffic class with goals to maximize network utilization and avoid starvation of low priority flows. The proposal incorporates the congestion detection and control algorithm. Upon detection of congestion, the congested node makes an estimate of the data rate that should be used by the node itself and its one-hop away upstream nodes. While estimating the data rate, the congested node considers the characteristics of different traffic classes along with their total bandwidth usage. The architecture uses a shared database to enable cross-layer interactions. Application's network protocol settings and the interaction with the shared database is done through a cross-layer optimization middleware.

Keywords: WSN, Wireless multimedia sensor networks, QoS, Cross- layer architecture, service differentiation

A preliminary version of this paper appeared in IWCMC 2011, July 5-8, Istanbul, Turkey. This version includes discussion on QoS metrics relevant to WMSNs, cross-layer architecture for multihop WMSNs, support for service differentiation, admission control, differentiated services based congestion control algorithm and detailed analytical evaluations of the proposed algorithms.

DOI: 10.3837/tiis.2012.01.011

1. Introduction

In recent years, Wireless Sensor Networks (WSNs) .[1] have gained attention in the research community due to their application in a multitude of real world situations. Invariably, the application domain imposes a restriction on the size and the cost of a single sensor node. These restrictions result in sensor nodes that are seriously constrained in terms of resources (CPU, memory, etc.). The decreasing cost of hardware such as CMOS cameras and microphones has resulted in a new variant of WSNs called Wireless Multimedia Sensor Networks (WMSNs) [2]. In WMSNs, sensor nodes are capable of capturing and communicating audio and video streams over a wireless channel. WMSN nodes are sophisticated compared to ordinary sensor nodes but still have limited resources.

Primarily, WSNs and WMSNs are deployed to capture and transmit information ubiquitously to sink nodes. Hence, the communication protocol plays a pivotal role for correct functionality of such networks. Scarce resources and the wireless communication medium inhibit the use of a traditional layered architecture, such as the TCP/IP protocol stack in WSNs [3]. TCP was originally designed for wired networks and its performance in wireless communication is reported to be poor [4]. Moreover, TCP is not well suited for multimedia flows due to its flow and congestion control mechanisms. UDP can serve as an alternative for TCP for multimedia applications, but it does not provide feedback about the status of the network that may be required for proper transmission of multimedia data. Hence, both TCP and UDP are not ideal transport layer protocols for WMSNs.

Cross-layer architecture design is emerging as an efficient technique for networking using wireless communication. In a cross-layer design, depending upon the condition of a wireless link, the MAC layer can choose appropriate error coding techniques. Similarly, the network layer can choose a path by taking input from the application and the physical layers. A cross-layer architecture can adapt the behavior of the protocol stack to the requirements of the application or, in the reverse direction, it can adapt the behavior of an application to the physical link conditions. To date, it has been assumed that WSNs run only a single application on a sensor node. As a result, researchers have developed cross-layer architectures that can work efficiently for single application-based wireless sensor networks.

In this paper, we argue that WMSNs are capable of transmitting audio and video streams along with scalar data. Therefore, there is a need for a cross-layer architecture that can support multiple communicating applications on a sensor node along with QoS provisioning support. Our proposed cross-layer architecture for QoS provisioning provides cross-layer interaction through a shared database. To support multiple applications, a separate space is reserved in memory where application-specific parameter settings for different layers of the protocol stack are stored through a Cross-Layer Optimization Middleware (CLOM). Application-specific settings are applied to the data while it is processed at different layers of the protocol stack. CLOM facilitates the tradeoffs in parameter settings depending upon the application requirements and physical network conditions.

The remainder of this paper is organized as follows. Related work is presented in Section 2. A novel cross-layer architecture for clustered and multi-hop based WMSNs along with an appropriate admission control mechanism and analytical results are introduced in Section 3. Then, a differentiated services based congestion control algorithm with experimental results is presented in Section 4. Finally, we conclude the paper in Section 5.

2. Related Work

Recently, cross-layer protocol design [5], [6] has gained momentum in wireless networks. Adaptability at different layers of the protocol stack w.r.t. network conditions and application requirements is the main advantage of cross-layer design. Cross-layer interaction results in an architectural design that is less modular than a traditional layered architecture. Fundamental goals of cross-layer design in WSNs are: light weight protocol stack, adaptability of protocols at different layers w.r.t. application requirements and physical channel conditions, reduced communication overhead for cross-layer interaction, and compatibility with traditional layered architectures. The guidelines for a cross-layer optimization framework for WSNs are given in [7].

2.1 Cross-layer Architectures for WSNs

In [8], the Cubic Cross-Layer (CCL) architecture for WSNs is presented. Cross-layering is achieved through a notification service and adaptation of the protocols to application specific needs. The Sensor Service Protocol (SSP) provides services that are rich enough to support a multitude of applications. Furthermore, the interfaces provided by SSP are platform-independent. CCL works with hierarchical clustered sensor networks. MAC, network, and transport layers are merged into one layer called Sensor Service Layer (SSL).

In [9], a cross-layer protocol for WSNs is presented. Functionalities of the transport, networking, MAC, and PHY layers have been merged into one single layer in order to gain energy efficiency. The protocol is composed of three features: initiative decision, receiver contention, and local cross-layer congestion control. In the initiative decision phase, channel conditions and local congestion status of those nodes who received an RTS packet are checked. In the receiver contention phase, one node is selected as the forwarding node. This decision is made depending upon the distance of a node from the sink as well as from the source node. Cross-layer congestion control is used to eliminate the congestion from the network. Since this protocol merges all layers into one, it is not compatible with a layered architecture. In [10], the X-Lisa architecture for WSNs is presented. A shared database is used for communication between different layers, containing three data structures: neighbor table, message pool, and sink table. In the neighbor table, information pertaining to different layers at neighboring nodes is stored. The message pool contains all the messages that are being sent and received. The sink table keeps track of all the sink nodes present in the network. A Cross-Layer Optimization Interface (CLOI) is used to exchange cross-layer information among different layers of the stack. The shared database is a novel and promising idea, but it increases memory requirements and computational cost. The continuous collection of information about neighbors requires additional bandwidth and memory. Moreover, this architecture does not support multiple applications on a single sensor node. TCLA [11] is a networking architecture for WSNs. TCLA uses the OSI model as a reference and tries to merge adjacent layers among which communication overhead is high, into virtual functional modules. These functional models are incorporated with agents that enable cross- layer interactions. TCLA merges application, presentation, and session layers into the AppM functional module. Transport and networking layers are merged into the NetM functional module. Data link and PHY layers are merged into the LinkM functional module. All modules have their own agents, disseminating management-related information to different layers. Each functional agent has a communication interface with other functional modules. The memory and energy requirements for TCLA are high.

In [12], a cross-layer architectural framework for WSNs is presented. This architectural framework proposes two distinct models. In the physical layer centered cross-layer network model, direct communication links are present between the PHY and the other four layers of the protocol stack. The reason for keeping the PHY at the center of the architecture design is that each layer adapts its behavior according to the physical channel conditions. Secondly, this arrangement facilitates keeping the layered architecture intact. In the application layer centered cross-layered network model, the application layer has bi-directional communication links with the remaining four layers. The application layer, being at the center of the architecture design, helps other layers to adapt to application requirements.

In [13], a cross-layer interaction scheme between network and MAC layer is presented to prolong the lifetime of a WSN. A distributed Time Division Multiple Access (TDMA) MAC protocol is proposed along with a reactive routing protocol for mobile WSN. The MAC operates in three states: active, passive, and dormant. In active state, a node can receive and initiate communication. In passive state, a node can request active nodes to relay its data. In dormant state, a node switches off its transceiver to operate in low power mode. Every node periodically advertises its time slot; this helps other nodes to select their time slot. The proposed routing protocol supports mobility in a WSN. Feedback from the MAC layer helps the networking layer to route efficiently. **Table 1** summarizes the reviewed architectures in terms of energy consumption, protocol stack size, cross-layer communication overhead, adaptability to channel conditions, and compatibility with a layered architecture. The classification of energy consumption was achieved by considering i) the overhead of the proposed architecture in terms of resource requirements and communication, and ii) the duty cycle requirement of the proposed architectures. An ideal architecture design results in network protocols with low to moderate energy consumption, moderate memory requirement, low to moderate cross-layer communication overhead, adaptability to channel conditions, and compatibility with the layered architecture. In current state-of-the-art, not a single architecture exhibits all these characteristics. Some adapt to the channel conditions and have low energy consumption but their memory requirements are high and they are not compatible with the layered architecture.

2.2 Cross-layer Architectures for WMSNs

Quality of Service (QoS) in WSNs usually refers to reliability, real-time transmission of delay sensitive data, keeping delay and jitter upper-bounded for multimedia real-time streams, and reserving bandwidth for multimedia applications. Recently, a substantial amount of research papers appeared on cross layer architecture design for WMSNs. In this research paper, we have surveyed the important work related to your proposal. A comprehensive survey on cross layer architectures can be found in [14].

In [15], a layer-less architecture is presented to support protocol-independent QoS. This architecture assumes that there are different routing and MAC layer protocols available on a node. At run time, this architecture can choose a particular protocol depending upon the QoS requirements of a packet. A QoS monitoring module keeps track of a node’s own resources, its neighbor’s resources, and network wide resources to facilitate selecting the appropriate protocol. An application/user QoS requirements module solicits the user requirements. The proposed QoS management framework is responsible for mapping the user requirements into the system priority classes. QoS polices decide the scheduling of packets and select the protocol to use for transmission. This architecture recommends the use of a single system-wide queue to obtain a global view of the system.

Table 1. Comparison of cross-layer architectures for WSNs

Metrics / Architecture	Energy Consumption	Small Protocol Stack Size	Cross-Layer Communication Overhead	Adaptability to Channel Conditions	Compatibility with the Layered Architecture
Distributed Efficient Architecture for WSNs [8]	Low	Moderately tiny	Moderate	Yes	No
Cross-Layer Protocol for WSNs [9]	Low	Yes	Low	Yes	No
Information Sharing Protocol Architecture for Sensor Networks [10]	High	No	Moderate	Yes	Yes
TCLA: Triangular Cross-Layer Architecture for WSNs [11]	High	No	High	Yes	Yes
A Central Networked Cross-Layer Design Framework for WSNs [12]	High	No	High	Yes	Yes
Prolonging the Lifetime of WSNs with Cross-Layer Interaction [13]	High	No	Low	No	Yes

In [16], a framework, called AMoQoSA, that can handle the heterogeneous nature (i.e., sensor nodes that differ in capabilities) of future WSNs is presented. The AMoQoSA framework enables the network to adaptively change its QoS behavior in order to continuously deliver QoS guarantees with respect to energy considerations. Three QoS profiles are recommended based on the capabilities of a sensor node: light weight QoS profile, basic QoS profile, and advanced QoS profile. The light weight profile is the simplest one. Nodes do not exchange information with other nodes but can receive other node's information. Its functionality is restricted to altering the packet parameters e.g., priority of a packet. The basic QoS profile nodes can exchange information with other nodes and can also alter the metrics used by different protocols. In the advanced QoS profile, nodes act as decision makers. Such a node can collect network-wide information and can instruct the nodes to change protocols at different layers.

In [17], a model for service differentiation in WMSNs is presented. The proposed model is based upon the differentiated services architecture with four traffic classes: Expedited Forwarding (EF), Assured Forwarding 1 (AF1), Assured Forwarding 2 (AF2), and Assured Forwarding 3 (AF3). For the EF class, priority scheduling is used and for the three other traffic classes, weighted round robin scheduling is used. Real-time traffic flows use EF, AF1 is used by high priority non real-time flows, AF2 is used for medium priority non real-time flow, and AF3 is used for low priority non real-time flows. A node's priority is calculated by summing up the priorities of its traffic flows. Global priority of a flow is calculated by summing up a node's own priority and the priority of its direct child nodes.

In [18], architectural and operational challenges for handling QoS traffic in WSNs are presented. Bandwidth limitations, removal of redundancy, energy and delay tradeoff, buffer size limitations, and support for multiple traffic types are pointed out as real challenges for providing QoS in WSNs. Furthermore, network dynamics, node deployment, data delivery model, node capabilities, and data aggregation/fusion are listed as design challenges for WSNs.

In [19], a cross-layer optimization framework for WMSNs is presented. The main goals are: (i) maximize data gathering from WMSN at the base station, (ii) minimize the delay, and (iii) predict the expected network lifetime. The architecture enables interaction among physical, network, and transport layers. To maximize the data gathering at the base station, the physical layer dynamically adjusts the data generation rate as well as the transmission radius. The network layer finds a set of node-disjoint multiple paths to the base station and the transport layer selects the appropriate routing path to minimize the delay. The shortcoming of the architecture is that it only considers multimedia traffic inside the WMSNs, but in reality WMSNs can generate scalar data as well, therefore the architecture needs components to better handle multimedia data in the presence of scalar data.

In [20], an energy-efficient and inter-layer interaction based communication architecture for Wireless Video Sensor Networks (WVSNs) is presented. The architecture defines interactions among application, transport, and networking layers. Moreover, the architecture defines new protocols namely: video compression sub-application layer protocol, real-time and reliable transport layer protocol, packet dropping policy, and energy-efficient single-path routing protocol. The architecture relies on compression algorithms to reduce the bandwidth required to transport real-time multimedia. Transport and networking layers cooperate to minimze delay, and the networking layer finds path that minimize energy consumption. The drawback associated with this architecture is that data packets are forwarded depending upon the prioirty of the data packet set by the coding scheme, therefore the scheme lacks support for real-time scalar data.

Table 2 summaries and evaluates the discussed QoS based cross-layer architectures for WSNs. The state-of-the-art in cross-layer QoS architectural design supports class-based service. Providing flow-based service is not feasible in resource and bandwidth constraint networks such as WSNs. QoS is provisioned in terms of delay, but for WMSNs, the architecture must provide some mechanism to provision bandwidth for different traffic classes. It is pertinent to note that none of the discussed architectures provide support for admission control. Architectural complexities are high although this may be tolerated in future WMSNs. Compatibility with a layered architecture is not always preserved. An ideal cross-layer architecture for WMSNs should provision QoS in terms of delay and bandwidth to different service classes. Admission control and/or flow control must be done. Efforts should be made to keep the architecture compatible with the traditional layered architecture.

Table 2. Comparison of cross-layer QoS architectures for WSNs

Metrics / Architecture	Service Granularity	Used QoS Metrics	Admission Control	Architecture Complexity	Compatibility with the Layered Architecture
Supporting Protocol Independent Adaptive QoS in WSNs [15]	Class based	Delay	No	High resource requirements	No
AMoQoSA [16]	Class based	Delay	No	High resource requirements	No
A Model for Differentiated Services Support in WMSNs [17]	Class based	Efforts are made to provide service differentiation	No	Moderate resource requirements	Yes
Cross-Layer Optimization for Data Gathering in WMSNs within Expected Network Lifetime [19]	Neither class- nor flow-based. Authors assume that there is only multimedia data.	Delay	No	Moderate	Yes
EVO Architecture [20]	Based on the priority of the multimedia frame produced by the compression algorithm	Delay	No	Moderate	No (Architecture proposes specialized protocols at different layers)

2.3 QoS Metrics for Wireless Sensor Networks

In traditional data networks, QoS guarantees are either provided on a per-flow basis nor on an aggregate basis. QoS requirements for traditional data networks invariably arise for multimedia applications. Therefore, QoS metrics for such applications are guaranteed bandwidth, minimum delay and jitter. Resource reservation, admission control, and packet scheduling are the main tools used to meet the stated QoS requirements in traditional data networks.

The primary purpose of WSNs is to monitor a given area and transmit reports to corresponding sink node(s). Hence, WSNs introduce some new QoS metrics apart from the traditional ones. In WSNs, area coverage, reliability, bandwidth guarantees, delay and jitter are basic QoS parameters. With the emergence of WMSNs, guaranteed bandwidth, minimum delay and jitter are getting more important for QoS provisioning. In the remainder of this section, we elaborate upon these metrics.

2.3.1 Area Coverage

In WSNs, there are two viewpoints of coverage: worst and best case coverage [21]. Worst case coverage tries to quantify QoS by finding areas of lower observability from sensor nodes and detecting breach regions. In best case coverage, QoS is quantified by finding areas of high observability from sensors and identifying regions that provide the best coverage in the sensing field.

Area coverage also depends on the sensor deployment strategy. There are two standard deployment strategies: deterministic and random. Deterministic deployment requires human intervention but yields a deployment that provides coverage that suits the application and overall system requirements. In a random deployment strategy, sensors are deployed in the sensing field with minimum human intervention but such a deployment may not comply 100% with overall system requirements.

2.3.2 Reliability

Reliability in QoS means that a higher priority packet should get to the sink node with high probability. The probability of a packet reaching the sink node depends on the wireless link conditions, routing protocol, congestion control mechanisms, and scheduling algorithm used at intermediate nodes. Furthermore, different traffic flows have different reliability requirements. Some applications require end-to-end reliability, some applications may only require link-level reliability, and some may not require any reliability. In WSNs, we cannot use a TCP-like approach because it is a highly resource-intensive protocol. At the same time, there is a need for a light-weight end-to-end reliability mechanism. In WSNs, we can employ a hop-by-hop ACK mechanism. In this way, the number of retransmissions only depends on local channel conditions. But this scheme should be applied sagaciously as waiting for an ACK for transmitted packet requires buffering the packet till the ACK is received. Secondly, retransmission is only beneficial if the retransmitted packet gets to the sink node within the allocated time frame i.e., the time until the received information is useful at the sink node. Moreover, in WSNs with heterogeneous traffic types, an adaptive reliability mechanism is required i.e., a reliability mechanism that adapts its behaviors w.r.t. the application requirements.

2.3.3 Latency

Latency can be defined as the overall time required by a data packet to go from source to destination. In fact, different traffic flows have different latency requirements. From a QoS perspective, latency is the maximum delay that a given flow can tolerate. Therefore, for efficient functionality of the application, the network needs to meet these latency requirements. In WSNs, latency primarily depends on channel conditions and transmission delay. If we employ some reliability mechanism, latency also depends on the number of retransmissions. Furthermore, if a sensor node has buffering capabilities, then latency would also depend on queuing delay.

Scheduling algorithms can be used to reduce latency of higher priority packets. Moreover, routing protocols can also help to reduce the latency by relaying high priority packets on shorter routes, routes with minimum error rate, routes with less congestion, or by using multipath routing.

2.3.4 Bandwidth Guarantees

Bandwidth guarantees are becoming important in WMSNs. Multimedia applications require constant or variable reserved bandwidth for the duration of the transmission. For example, a sensor network deployed in a forest to detect fire may contain sensor nodes that capture real-time images and transmit them to the sink node. Such an application requires guaranteed bandwidth, yet providing bandwidth guarantees in WMSNs is not feasible at present. The way to deal with this problem is to prioritize such traffic over non real-time data. In cluster-based sensor networks, cluster heads can poll such sensor nodes frequently to provide soft bandwidth guarantees. Moreover, we need to segregate real-time data from non real-time data by incorporating priority scheduling.

2.3.5 Jitter

Jitter, which is the deviation from the mean delay, is an important QoS parameter in WMSNs. Video sensors send real-time images to the sink node. For proper visualization of the data, it is important that jitter remains within the prescribed limits. To circumvent the effects of jitter, we can use buffering at the sink nodes, but this can increase the playout latency, so we need to strike a balance. Buffering at sink nodes is a possible solution since the majority of sink nodes have significant resources, such as memory required for buffering.

3. Cross-Layer Architectures for QoS Provisioning in Wireless Multimedia Sensor Networks

In this section, we discuss our proposed cross-layer architecture for QoS provisioning in WMSNs. Our goal is to develop an architecture that has the following characteristics:

a) QoS provisioning support;
b) Interoperability with a layered architecture;
c) Moderate resource requirements on a sensor node;
d) Supporting network conditions and application requirements tradeoffs.

In the following sub-sections, we shall eloborate on different components of the proposed cross-layer architecture for WMSNs.

3.1 Supported Traffic Classes

The proposed architecture defines forwarding behavior w.r.t. the following traffic classes.

a) Real-time Loss In-tolerant Data: Such data invariably arises from time-critical monitoring processes. Since this is the highest priority class, data must be relayed to the sink node with high reliability and minimum delay. Given that the amount of critical data is relatively low and that they are short-lived, we assume that the bandwidth requirement in this traffic class is relatively small.
b) Real-time Loss Tolerant Data: Important scalar readings like an abrupt change in temperature at a particular instant in time or other important sensor readings must be relayed to the sink node immediately. Such data is loss-tolerant because of the dense deployment of senor nodes i.e., it is highly likely that if one reading is corrupted during transmission, the reading sent by a nearby sensor will compensate for it. Efforts are required to deliver data belonging to this traffic class with minimum delay.
c) Real-time Loss Tolerant Multimedia Streams: It is a well known fact that real-time multimedia streams can afford some packet loss but they are very sensitive to delay and jitter. Excessive delays can render real-time multimedia data useless, therefore efforts must be incorporated to transmit data with minimum possible delay. Depending on the current traffic load inside the network, bandwidth should be reserved for this traffic class.

d) Delay Tolerant and Loss Tolerant Multimedia Streams: There are multimedia streams that can be processed offline, such streams arise from environmental monitoring systems. These streams can tolerate delay as well as packet loss. Data belonging to this traffic class can be stored at a sensor node and can be transmitted when the network is experiencing low traffic load. There is a limit on the data storage capabilities of sensor nodes, and with current state-of-the-art nodes storing a lot of data may not be feasible.
e) Delay Tolerant and Loss In-tolerant Data: Such data arises from monitoring systems and typically emerges in response to a query that does not involve mission-critical aspects of the system.
f) Delay Tolerant and Loss Tolerant Data: Data that is neither mission critical nor time critical.

3.2 Cross-Layer Architecture for Cluster based WMSNs

In this section, we discuss the proposed cross-layer QoS provisioning architecture in the context of cluster based WMSNs.

3.2.1 Topology and Deployment Methodology

Real-time multimedia applications are bandwidth intensive, therefore we need to commit a certain amount of bandwidth to such applications. As we are dealing with heterogeneous traffic, we need to meet deadlines of real-time critical scalar data. Such real-time critical data has an upper bound on delay that must be met. Furthermore, we have data for which reliability is of utmost importance. Therefore, the architecture must provide mechanisms for reliability. Considering the stated goals, we need to devise a network topology that yields high bandwidth utilization with minimum interference.

To meet the stated goals, we have opted for a deterministic deployment methodology because we can deploy sensors according to the applications requirements. Secondly, control message overhead is low. We propose to partition the sensing fields into hexagonal cells and suggest using the seven cells frequency reuse pattern to minimize interference. To further minimize the interference and synchronization issues, we argue for separate sink node inside each cell. This may increase the cost of deploying a sensor network but yields better results. For an assessment of the number of sinks required, consider the example of a square sensing region of length 100 meters. To cover the square region, we further assume, for the sake of simplicity, that cells are of square shape and the length of each square is 14.14 meters. The transmission range of Zigbee or IEEE 802.15.4 is almost 10 meters, so a sink deployed in the centre of each cell can communicate with sensors deployed anywhere within the cell. With this setting, we require 50 cells to cover the whole region and therefore, 50 sink nodes would be required. The infrastructure cost is high but this yields many benefits because cells are of small size and nodes within small cells can utilize more bandwidth. In order to provide Internet connectivity, sink nodes are also equipped with at least one IEEE 802.11 interface and there is one access point within the transmission range of each sink node. This access point accommodates all sink nodes within its transmission range.

Each cell is using a separate frequency band and we are using the seven cell frequency reuse pattern, therefore 1/7 of the total bandwidth is available inside each cell. This deployment will yield less delay as each cell has its own sink node. Furthermore, more bandwidth is available to a smaller number of sensor nodes, resulting in enhanced throughput. This topology yields a more reliable system as sink nodes are adjacent to the sensing nodes.

3.2.2 Architecture Design

We argue that QoS provisioning in cluster based WMSNs can best be done with our described network setup and topology. All communications take place without intermediate nodes relaying data to the sink node. This eliminates the need for a network layer from the protocol stack on member sensor nodes, hence saving memory and eliminating control message overhead for finding a multi-hop path to the sink node. The proposed cross-layer architecture is shown in **Fig. 1** Applications set forth their requirements through a cross-layer optimization middleware. The cross-layer optimization middleware sets appropriate parameters at different layers of the protocol stack, keeping in mind the application requirements and the network conditions that are obtained through physical channel conditions and from MAC layer feedback. Furthermore, the middleware informs applications about the status of the network so that applications can adapt their behavior accordingly.

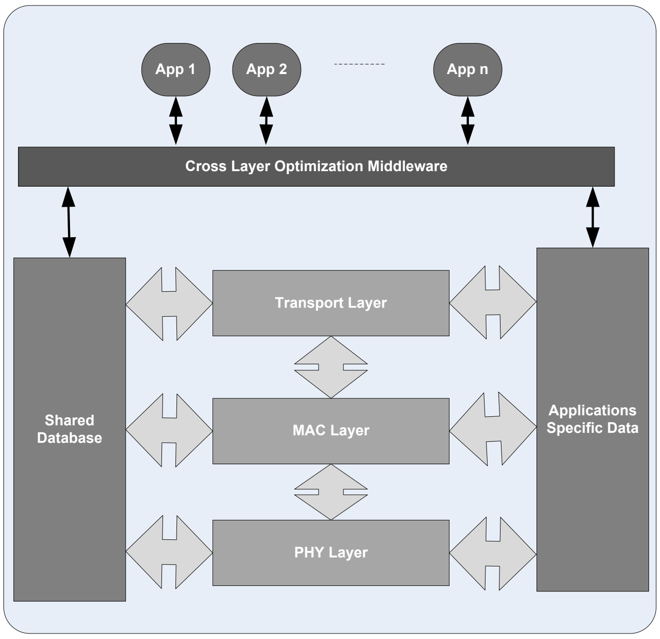

Fig. 1. Cross-layer QoS architecture for cluster based WMSNs

Each layer stores important information in a shared database so that different entities can access the information. The configuration parameters at different layers can only be set by the cross-layer optimization middleware. We have opted for a shared database mechanism to provide cross-layer interaction because it maintains the modularity of the layered architecture. Furthermore, it can be incorporated into existing protocol stacks with minimum effort. However, the drawbacks associated with this scheme are i) we need to implement mechanisms to keep the database in a consistent state through locking mechanism, and ii) we require additional memory to store application parameters corresponding to the transport, MAC, and PHY layers of the protocol stack. In fact, the memory requirement increases with respect to the number of applications running on a sensor node.

To the best of the authors' knowledge, there does not yet exist any architecture for QoS provisioning in WMSNs that provides support for multiple applications. We argue that future multimedia wireless sensor nodes will be capable of running multiple applications. Therefore, we have provided support for multiple applications in our architectures. Different applications will have different requirements and consequently may want to configure the network protocol parameters according to their own needs. For this purpose, a separate storage area is reserved for each application. For example, it is possible that one application requires a particular Forward Error Correction (FEC) technique and another requires a different FEC technique. We further assume that, before selecting a particular FEC scheme, the sensor nodes used some capabilities exchange protocol to figure out supported FEC schemes at the other sensor node. Each layer discriminates packets from different applications and makes such decisions based on the specific parameters stored in the application-specific storage area.

3.2.3 Bandwidth Assignment and Admission Control Mechanisms

For the design of bandwidth assignment and admission control mechanims, we assume that the cluster head (CH) is able to estimate the available bandwidth referred to as β_{avb}^{CH} .

The first two traffic classes carry critical scalar data, therefore mechanisms must be in place to transmit such data with highest priority and reliability. Data pertaining to these first two traffic classes emerge infrequently and are short lived, therefore the average bandwidth consumption attributeable to these two traffic classes is not too high. We denote the mean bandwidth consumption of these two traffic classes as $\beta_1^{CH} + \beta_2^{CH}$. The following equation gives the remaining bandwidth at the cluster head.

$$\beta_{rem}^{CH} = \beta_{avb}^{CH} - \left(\beta_1^{CH} + \beta_2^{CH}\right) \tag{1}$$

We have devised the following linear program to allocate a portion of available bandwidth to the member nodes.

$$Maximize \sum_{i=1}^{\omega} \tau i \beta i \tag{2}$$

s.t.

$$\chi^{i}_{min} \leq \beta i \leq \chi^{i}_{max} \qquad i \in \omega \tag{3}$$

$$\sum_{i=1}^{\omega} \beta i = \beta_{rem}^{CH} \qquad i \in \omega \tag{4}$$

Fig. 2. Linear program for allocating bandwidth to member nodes

In the above linear program, ω represents the number of member nodes requesting bandwidth. χ^{i}_{min} represents the minimum threshold bandwidth associated with each requesting node (the minimum bandwidth value for each member node is calculated based upon the requested bandwidth as well as the bandwidth available at the CH) and χ_{max} represents the maximum bandwidth that can be assigned to a node. β_i is the decision variable that gives the optimal value of the bandwidth for member node i. τ_i is the priority index of each node ($0 \leq \tau_i \leq 1$). The value of τ_i for each member node is derived from the priorities of the traffic originating from the member node. For example, if a node wants β_3, β_4, β_5, and β_6 amounts of bandwidth in classes 3 through 6 respectively, then τ_i is calculated as $((\beta_3 \, x \, \rho_3) + (\beta_4 \, x \, \rho_4) + (\beta_5 \, x \, \rho_5) + (\beta_6 \, x \, \rho_6))/ \beta^{i}_{req}$. Here β^{i}_{req} represents the total bandwidth requested by a node i. ρ_i represents the priority of traffic class i. It is left to the discretion of the network designer to assign priorities to the different traffic classes. The value of χ_{min} is determined as shown in Fig. 3.

$$\beta^{i}_{u} = \frac{\beta^{CH}_{avb}}{\omega}$$
$$\chi^{i}_{min} = \beta^{i}_{u} x\ \tau_{i}$$
$$If(\beta^{i}_{req} \leq \chi^{i}_{min})$$
$$\chi^{i}_{min} = \beta^{i}_{req}$$

Fig. 3. Procedure for calculating the value of χ^{i}_{min}

Based upon the bandwidth allocated to each member node, the CH issues a Time Division Multiple Access (TDMA) schedule for member nodes. The bandwidth assignment procedure and TDMA schedule collectively acts as an admission control procedure.

3.3 Cross-Layer Architecture for Multihop WMSNs

In this section, we discuss the proposed cross-layer QoS provisioning architecture in the context of multihop WMSNs. While we believe the clustered deployment to be a good approach, it might not be possible for the clusters themselves to be directly connected to a central gateway. In this case, the clusters need to reach the base stataion using multiple hops.

3.3.1 Topology and Deployment Methodology

In a multihop WMSN, sensor nodes colloborate in a multihop manner to send data to the sink node as shown in **Fig. 4**. It depends on the phenomenon under observation to decide on the sensors deployment methodology.

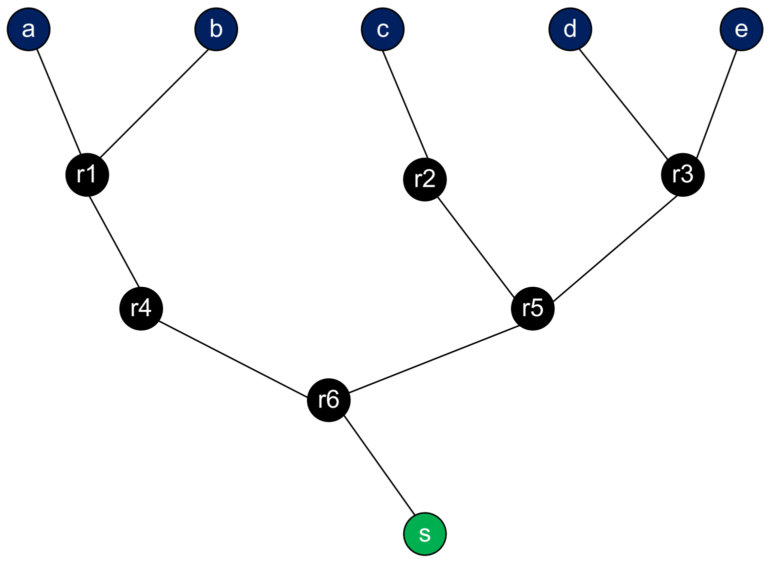

Fig. 4. Multihop WMSN

3.3.2 Architecture Design

Fig. 5 shows the cross-layer architecture design for multihop WMSNs. Since there is a need for a routing protocol to send data to the base station, we have included the networking layer in our proposed architecture (which is different from the architecture we proposed for cluster based WMSNs in **Fig. 1**). The networking layer can have implementations of different routing protocols that suit the requirements of different applications. For example, to meet the QoS requirements of real-time multimedia application, the routing layer should contain an implementation of a routing protocol that can find paths which are able to meet the QoS requirements of a flow.

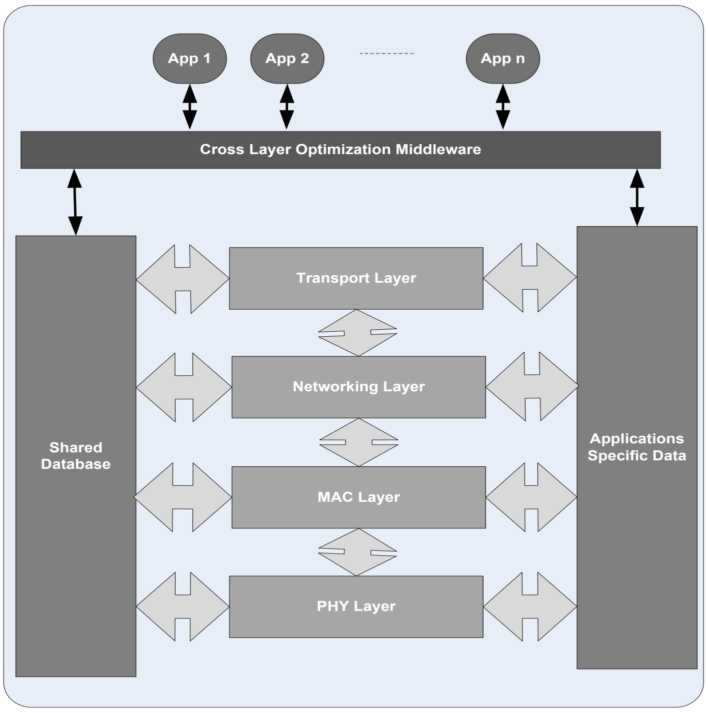

Fig. 5. Cross-layer QoS architecture for multihop WMSNs

3.3.3 Bandwidth Assignment and Admission Control Mechanisms

To estimate the share of bandwidth allocated to each traffic class in multihop WMSNs, we need to consider the stated charateristics of each traffic class. We made the following assumptions. Total bandwidth available at a particular node '*n*' is β^{n}_{avb}. Traffic that is mapped to each traffic class is being generated with a Poisson source with mean arrival rates of $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5,$ and λ_6. The service rate that each traffic class should receive is denoted by $\mu_1, \mu_2, \mu_3, \mu_4, \mu_5,$ and μ_6, where $0 \le \mu_i \le 1$. A particular μ_i assumes a value equal to 0 when there is no flow corresponding to traffic class '*i*'. A particular μ_i assumes a value of 1 when there are no flows pertaining to other traffic classes and class '*i*' has enough flows to completely consume the bandwidth. While assigning service rates, the following equation must hold.

$$\sum_{i=1}^{6} \mu_i \beta^{n}_{avb} \le \beta^{n}_{avb} \;\forall n \tag{5}$$

In (5), $\mu_i \beta^{n}_{avb}$ represents the total bandwidth assigned to a particular traffic class '*i*'. Hence, the above equation ensures that the bandwidth allocation to all the traffic classes must be less than or equal to the total bandwidth available at a particular node.

Let us analyze the bandwidth assignment procedure to each traffic class depending upon the requested bandwidth and priority of the traffic class. Assuming that the average bandwidth consumption in the first two classes is ($\beta_1^n + \beta_2^n$), the remaining bandwidth for all other classes is:

$$\beta_{rem}^n = \beta_{avb}^n - (\beta_1^n + \beta_2^n) \quad \forall n \tag{6}$$

The first two traffic classes get the bandwidth allocated as per their requirements (assuming data pertaining to these traffic classes are short and emerges rarely). Bandwidth consumption in these first two traffic classes is monitored and is averaged out to calculate the values of $\beta^n{}_1$ and $\beta^n{}_2$. We have devised the following linear program to allocate a portion of the available bandwidth to the remaining traffic classes. The linear program allocates bandwidth for a constant time interval *T*. *T* should be large enough to avoid redundant calculations and at the same time it should be small enough so that changes in the bandwidth requirements can be tracked appropriately.

$$Maximize \sum_{i=3}^{6} \rho_i \beta_i \tag{7}$$

s.t.

$$\beta_i \geq \xi_i \qquad \forall_{3 \leq i \leq 6} \tag{8}$$

$$\beta_i \leq \Psi_i \qquad \forall_{3 \leq i \leq 6} \tag{9}$$

$$\sum_{i=3}^{6} \beta_i = \beta_{rem}^n \tag{10}$$

Fig. 6. Linear program for assigning the bandwidth to the data traffic classes

In the above linear program, ρ_i and β_i represent the priority index and the bandwidth required in the i^{th} traffic class respectively. The value of ρ_i ranges between 0 and 1. It is left to the discretion of the system administrator to assign values of ρ_i depending upon the kind of phenomenon under observation and the ultimate purpose for deploying the network. ξ_i denotes the lower threshold value for each traffic class and this value can vary depending upon the characteristics of different priority classes as well as the mean arrival rate of traffic in the system. Ψi denotes the upper threshold value of a service rate that can be allocated to traffic class *i*. Note that the constraint in Equation (8) requires that each traffic class obtains a share from the available bandwidth to avoid starvation. It is possible that, at any given instance, there is no flow pertaining to a certain traffic class. In this case, the above linear program excludes that traffic class from its calculations.

Please note that the linear program given in **Fig. 6** can also be used by the member nodes of a cluster to allocate the assigned bandwidth to different traffic classes.

3.4 Analytical Results

In this section, we present an example to demonstrate the application of our proposed bandwidth assignment and admission control mechanisms. We used TORA [22] for solving the linear programs. The following data set (**Table 3**) is used to allocate a portion of the bandwidth to sensor nodes. We apply the proposed solution to three communication paradigms: single hop WMSN, multihop WMSN, and clustered WMSN. For estimating the available bandwidth, the algorithm presented in [23] can be used.

Table 3. Data set for evaluations

Parameter	Value
System bandwidth (β_{total})	250 Kbps
Available bandwidth as given by bandwidth estimation algorithm. β^{i}_{avb}.	40 Kbps
Bandwidth required for real-time loss tolerant multimedia stream (β_3)	25 Kbps
Bandwidth required for delay tolerant and loss tolerant multimedia stream (β_4).	20 Kbps
Bandwidth required for delay tolerant and loss in-tolerant data (β_5).	5 Kbps
Bandwidth required for delay tolerant and loss tolerant data (β_6).	5 Kbps
Bandwidth required by real-time loss in-tolerant data and real-time loss tolerant data ($\beta_1 + \beta_2$).	5 Kbps
Total bandwidth available after reserving ($\beta_1 + \beta_2$) = β^{i}_{avb} - ($\beta_1 + \beta_2$).	35 Kbps

3.4.1 Bandwidth Assignment and Admission Control Evaluation in Single hop WMSNs

From **Table 3**, 35 Kbps of bandwidth is available to a node after subtracting ($\beta_1 + \beta_2$) i.e., the average bandwidth required by the first two traffic classes. The total bandwidth required for the remaining traffic classes is 25, 20, 5, and 5 Kbps, for a total of 55 Kbps. As we can see, the bandwidth required by the node is greater than the available bandwidth, therefore we need to allocate bandwidth using the linear program shown in **Fig. 6**. In this example, we have given more importance to the real-time loss tolerant multimedia class and the delay tolerant and loss in-tolerant traffic class. Therefore, $\rho_3 = 1$, $\rho_4 = 0.4$, $\rho_5 = 0.7$, and $\rho_6 = 0.2$. The value of $\xi_i = (\rho_i \, x \, \beta_{avb})/n$. The value of n is 4 because we are considering four traffic classes. If the value of ξ_i exceeds the maximum bandwidth required, then ξ_i is set equal to the maximum. Similarly, $\psi_i = \beta_i$. Therefore, $\xi_3 = (\rho_3 \, x \, \beta_{avb})/4 = 8.75$ Kbps. $\xi_4 = (\rho_4 \, x \, \beta_{avb})/4 = 3.5$ Kbps. $\xi_5 = (\rho_5 \, x \, \beta_{avb})/4 = 6.125$ Kbps (more than required, therefore $\xi_5 = 5$ Kbps). $\xi_6 = (\rho_6 \, x \, \beta_{avb})/4 = 1.75$ Kbps. $\psi_i = \beta_i$, therefore ψ_3, ψ_4, ψ_5, and ψ_6 are 25 Kbps, 20 Kbps, 5 Kbps, and 5 Kbps respectively.

Using the given data, we solved the linear program given in **Fig. 6**. The objective function value and the bandwidth assigned to each traffic class are shown in the following table.

Table 4. Bandwidth assigned to the traffic classes

Objective Function Value	Class 3	Class 4	Class 5	Class 6
666,000,000	24.75 Kbps	3.5 Kbps	5 Kbps	1.75 Kbps

The results shown in **Table 4** suggest that our linear program assigns bandwidth according to the expectations of the network designer. As we have used a priority index that favors class 3 and class 5 traffic flows, our linear program allocates almost the requested bandwidth to these classes. At the same time, it also prevents other traffic classes from starvation. It is worth noting that class 4 gets 3.5 Kbps but it has made a request for 20 Kbps. This is due to the fact that the priority index for this traffic class is small, which may be due to the fact that such data can be buffered and transferred later.

3.4.2 Bandwidth Assignment and Admission Control Evaluation in Multihop WMSNs

The performance of the bandwidth assignment and admission control mechanism for WMSN in the multi-hop based forwarding paradigm depends upon the accuracy of the algorithm used for estimating the available bandwidth. If we suppose that the estimated available bandwidth is the same as given in **Table 3**, then the results will be similar to those shown in **Table 4**.

3.4.3 Bandwidth Assignment and Admission Control Evalution in Clustered WMSNs

Here we assume that the number of member nodes inside each cluster is four and that the total available bandwidth is 40 Kbps. After subtracting the average bandwidth consumption of the first two traffic classes ($\beta_1 + \beta_2$), the available bandwidth is 35 Kbps The following table lists the nodes' required bandwidth in different classes and their corresponding τ_i values.

Table 5. Bandwidth requirement and τi

Node	Class 1	Class 2	Class 3	Class 4	Required Bandwidth	τ_i
1	10 Kbps	5 Kbps	5 Kbps	0 Kbps	20 Kbps	0.7750
2	8 Kbps	5 Kbps	0 Kbps	2 Kbps	15 Kbps	0.6933
3	20 Kbps	0 Kbps	0 Kbps	0 Kbps	20 Kbps	1.0000
4	0 Kbps	20 Kbps	0 Kbps	5 Kbps	25 Kbps	0.3600

Table 6 shows the value of χ_{min} calculated as per the procedure given in **Fig. 3**.

Table 6. χmin values of member nodes

Node 1	Node 2	Node 3	Node 4
6.781 Kbps	6.066 Kbps	8.75 Kbps	3.15 Kbps

The value of χ^i_{max} corresponding to each member node *i* is set equal to the maximum bandwidth requested by node *i, i.e.*, $\chi^i_{max} = \beta^i_{req}$. It is evident from **Table 7** that our linear program allocates more bandwidth to the nodes having higher priority index. Moreover, it allocates considerable bandwidth to other nodes so that they do not starve.

Table 7. Bandwidth assigned to member nodes

Objective Function Value	Node 1	Node 2	Node 3	Node 4
106520	6.78 Kbps	6.07 Kbps	19.0 Kbps	3.15 Kbps

Afterwards, each node uses the linear program presented in **Fig. 6** to calculate the service rate that must be assigned to traffic pertaining to different classes. The network designer sets priorities of different traffic classes depending upon the purpose of the deployed network.

4. Differentiated Services based Congestion Control Algorithm

In this section, we propose a differentiated services based congestion control algorithm that can be integrated with the proposed cross-layer architecture for WMSNs.

Congestion in WMSNs degrades the performance of traffic flows present in the network. Possible causes of congestion in WMSNs include: occurrence of a critical event, excessive event reporting, multimedia data, and hot spots. The consequences of congestion are: decreased reliability, increased delay and jitter, and wastage of resources (i.e., bandwidth and energy). Therefore, without having congestion detection and control mechanisms in place, meeting QoS requirements for inelastic applications becomes a daunting task.

Keeping in mind the consequences of congestion in WMSNs, we propose a differentiated services based congestion control algorithm for WMSNs. Therefore, the proposed algorithm assumes that WMSN generates traffic in the defined traffic classes. Upon detecting the congestion, the congested node in the network optimally calculates the reduced share of the bandwidth for all one hop away upstream nodes for which it is acting as a relaying node. The proposed solution takes a different and novel approach towards congestion control, compared to contemporary congestion control algorithms for WMSNs. The proposed solution not only informs upstream nodes about the congestion, but it also informs the upstream nodes about the modified data rate that they should use to alleviate the congestion. To decide the data rate pertaining to upstream nodes, the algorithm considers the characteristics of multimedia data and wireless communication. Upstream nodes producing less data and low priority data are more penalized compared to the nodes producing the bulk volume of data and high priority data. The rational behind this design is twofold: *i)* multimedia data invariably occurs in large quantity and *ii)* receiving more data from a particular upstream node can be taken as an indication of good wireless channel condition (from the source node to that particular upstream node) considering that a fair MAC layer is in use. As a result, data originating from such areas must be penalized less.

Following are the components of the congestion control algorithm.

1) Congestion detection
2) Procedure for calculating bandwidth share for upstream nodes

We made the following assumptions for designing the congestion control algorithm:

1) Any intermediate node n knows the mean data arrival rate i.e., λ_i^j pertaining to traffic class i from upstream node j. Applications local to node n may generate traffic pertaining to class i, with the mean arrival rate of λ_i^n. Therefore, the mean arrival rate in a particular traffic class i, $\lambda_i = \left(\sum_{j=1}^{\omega} \lambda_i^j\right) + \lambda_i^n$, where ω represents the one hop away upstream nodes (for which a node is acting as a relaying node).
2) Each node estimates the available bandwidth β_{avb} by monitoring the wireless channel or through other available bandwidth estimation algorithms for wireless networks.

4.1 Congestion Detection

As each node knows the bandwidth available to it, each node monitors λ_i and mean service rate μ_i corresponding to each traffic class i. Whenever the set $\{\lambda_i > \mu_i | 1 \leq i \leq 6\} \neq 0$, a node starts to monitor queue(s) corresponding to traffic class(es) for which $\lambda_i > \mu_i$. When the queue occupancy reaches a threshold level τ_i pertaining to traffic class i, the node starts the congestion control mechanism. The following subsection eleborates the proposed congestion control mechanism. We define the mean arrival rate as $\lambda = \vartheta/\varepsilon$, where ϑ is the flows' mean data generation rate and ε is the system's total data rate. Similarly, $\mu = {}^{\varrho}/_{\varepsilon}$, where ϱ is the mean service rate in terms of number of bits transfered per unit time.

4.2 Procedure for Calculating Bandwidth Share for Upstream Nodes

When congestion is detected in the network, the system starts an algorithm to calculate the modified share of the bandwidth for each one hop away upstream node, given that the node n is acting as a relaying node for that particular one hop away upstream node. If the congested node is generating some flows locally then the algorithm also calculates the modified share of bandwidth for the congested node. The linear program shown in **Fig. 7** assigns the modified bandwidth to each one hop away upstream node. While optimizing the bandwidth assignment procedure, the linear program considers the priority of the node to whom bandwidth will be assigned. The modified bandwidth information is sent to all concerned one hop away upstream nodes and to the node itself. Nodes can then reduce transmission rates for each traffic class depending on the available bandwidth, priority of the traffic class, and amount of data generated in each traffic class. If the receiving node also experiences congestion, then the process continues until the source of the congestion gets informed.

$$Maximize \sum_{j=0}^{\omega} \eta_j \beta_j \tag{11}$$

s.t.

$$\phi_j^{min} \leq \beta_j \leq \phi_j^{max} \qquad \forall j \tag{12}$$

$$\sum_{j=0}^{\omega} \beta_j \leq \beta_{avb}^{n} \tag{13}$$

Fig. 7. Linear program for rate adjustment

In the above linear program, β_j is the decision variable i.e., bandwidth that will be assigned to the j^{th} node, η_j represents the priority of the j^{th} node and $0 \leq \eta_j \leq 1$. ϕ_j^{min} and ϕ_j^{max} give the minimum and maximum bandwidth that can be assigned to the j^{th} node, and Equation (13) ensures that the total assigned bandwidth does not exceed the available bandwidth. Please note, $j = 0$ represents the congested node and its rate is adjusted if the congested node is generating flows locally. The methods for calculting different parameters used in the above linear program are discussed below.

The value of η_j for each node is derived from the priorities of the flows originating from node *j*. If the mean bandwidth utilized by node *j* in the traffic class *i* is $\overline{\beta_i^j}$ then the mean bandwidth utilized by node *j* is $\overline{\beta_{tot}^j} = \sum_{i=1}^{6} \overline{\beta_i^j}$. The value of η_j is calculated as follows.

$$\eta_j = \frac{\sum_{i=1}^{6}\left(\alpha_i \times \overline{\beta_i^j}\right)}{\beta_{avb}^{n}} \tag{14}$$

In Equation (14), α_i represents the priority of traffic class *i*. β_{avb}^{n} represents the bandwidth available at the congested node. To calculate the value of ϕ_j^{min}, the algorithm first calculates the value of β_u^j i.e., the bandwidth that could be assigned to node *j*, if node *n* distributes β_{avb}^{n} uniformly to the nodes $\left(\beta_u^j = \frac{\beta_{avb}^{n}}{\omega + k}\right)$. The value of *k* is 1 if the congested node is generating flows locally and 0 otherwise. Afterwards, ϕ_j^{min} is set equal to $\beta_u^j \times \eta_j$. If $\overline{\beta_{tot}^j} < \phi_j^{min}$ then $\phi_j^{min} = \overline{\beta_{tot}^j}$. The value of ϕ_j^{max} is always set equal to $\overline{\beta_{tot}^j}$. Fig. 8 shows the pseudo-code of the congestion control algorithm.

```
Algorithm 1: Congestion Detection and Control
Data: $\lambda_i$[ ],$\alpha_i$[ ], $\mu_i$[ ], $\overline{\beta_i^j}$[ ], $\overline{\beta_i^n}$[ ], $\overline{\beta_{tot}^i}$[ ], $\beta_{avb}^n, \omega$
Result: $\beta$[ ]
1  $isCongestion \leftarrow 0$;
2  if $\forall\lambda_i\exists(\lambda_i > \mu_i)$ then
3      $isCongestion \leftarrow monitorQueue(i[\ ])$;
4  end
5  if $isCongestion$ then
6      $\eta\ [\ ] \leftarrow calculateEta(\alpha_i[\ ]\ , \overline{\beta_i^j}[\ ], \overline{\beta_{tot}^i[\ ]})$;
7      $\phi_{min}[\ ] \leftarrow calculatePhiMin(\beta_{avb}^n, \omega, \overline{\beta_{tot}^i}[\ ])$;
8      $i \leftarrow 0$;
9      for $i \leq 6$ do
10         $\phi_{max}[i] \leftarrow \overline{\beta_j}$;
11     end
12     $\beta\ [\ ] \leftarrow reassignBandwidth(\phi_{min}[\ ], \phi_{max}[\ ], \eta\ [\ ])$;
13     $i = 0$;
14     for $i \leq |\omega|$ do
15         $communicateCongestion(\beta[i])$;
16         $i \leftarrow i + 1$;
17     end
18 end
```

Fig. 8. Pseudo-code for congestion control algorithm

4.3 Experimental Results

The network shown in **Fig. 4** is used to demonstrate the effectiveness of the proposed congestion control algorithm for WMSNs. Sensor nodes labelled *a*, *b*, *c*, *d*, and *e* generate traffic corresponding to traffic classes 1, 3, and 5. Sensor nodes start to generate data corresponding to different traffic classes as soon as they wake up.

T_{awk}^i denotes the wake up time of node *i*. Sensor node *a* uniformly generates 10 pkts/sec, 1 pkt/sec, and 10 pkts/sec with packet size of 500 bits, 500 bits, and 200 bits corresponding to traffic classes 3, 1, and 5 respectively. Sensor node *b* uniformly generates 10 pkts/sec each corresponding to traffic classes 3, and 5, with packet size of 400 bits and 150 bits respectively. Sensor node *c* uniformly generates 2 pkts/sec, and 20 pkts/sec with packet size of 500 bits and 200 bits corresponding to traffic classes 1 and 5 respectively. Sensor node *d* uniformly generates 30 pkts/sec and 10 pkts/sec with packet size of 400 bits and 150 bits corresponding to traffic classes 3 and 5 respectively. Sensor node *e* uniformly generates 20 pkts/sec each with packet size of 500 bits and 200 bits corresponding to traffic classes 3 and 5 respectively. **Table 8** shows the data generation rate and T_{awk}^i, pertaining to each source node.

Table 8. Nodes specific parameters

Node ID	Data Generation Rate	T^i_{awk}
a	7.5 kbps	0 sec
b	5.5 kbps	10 sec
c	5 kbps	20 sec
d	13.5 kbps	25 sec
e	14 kbps	25 sec

We assume a wireless network based on the IEEE 802.15.4 standard supporting a data rate of 40 kbps, therefore total system data rate (ε) is 40 kbps. From the network topology and data generation rates of different source nodes, it can be inferred that congestion is bound to happen at node *r6* at time 25 seconds. Therefore, the system detects congestion and activates the congestion control algorithm. **Table 9** shows different values as given by the algorithm. Please note, we have used α_1 = 0.6, α_3 = 0.5, and α_5 = 0.3 in our experiments. The value of β^n_{avb}= 40 kbps.

Table 9. Linear program parameters

Node ID	η_j	ϕ^{min}_j	ϕ^{max}_j
r4	0.146	2.925 kbps	13.000 kbps
r5	0.361	7.225 kbps	32.500 kbps

It can be observed from **Table 10** that the proposed congestion control algorithm favours nodes that produce the bulk of data with higher priorities. **Table 10** shows that node *r5* is allocated the needed bandwidth but reduced bandwidth is granted to *r4*. Previously *r4* was generating data at 13 kbps. Now, *r4* has been instructed to cut down the rate to 7.5 kbps. As *r4* is acting as a relaying node for nodes *a* and *b*, which are generating data at 7.5 and 5.5 kbps respectively, node *r4* starts the congestion control procedure.

Table 10. Bandwidth share

Node ID	β_j
r4	7.5 kbps
r5	32.5 kbps

Table 11 shows the final outcome of the congestion control algorithm. It shows the share of the bandwidth assigned to nodes *a* and *b*. Finally, congestion control information has reached the source nodes. Source nodes can penalize different flows depending upon their priority and data generation rate.

Table 11. Bandwidth share

Node ID	β_j
a	6.275 kbps
b	1.225 kbps

For the purpose of comparison, we have executed the rate adjustment algorithm presented in [24] on the network topology shown in **Fig. 4**. This algorithm determines the modified rate on the basis of the global priority associated with each node. The global priority of the node is the summation of the priority of each flow originating or passing through the node. Under the given traffic generation scenario, **Table 12** shows the global priorities of nodes *r4*, *r5* and *r6* as per the algorithm given in [24].

Table 12. Global priorities

Node ID	Global Priority
r4	2.2
r5	2.5
r6	4.7

Table 13 shows the modified bandwidth assigned to nodes *r4* and *r5*. It is interesting to note that this algorithm has assigned approximately the same amount of bandwidth to nodes *r4* and *r5*, regardless of the fact that node *r6* is receiving more data from node *r5*. This occurs because the rate adjustment algorithm only uses the number of flows and their priorities as a parameter. In this particular experimental setup, almost the same number of flows was using nodes *r4* and *r5* as their relaying nodes and the summation of their priorities were also almost equal, hence approximately the same amount of bandwidth is assigned to both upstream nodes. Since this algorithm does not take into account the data generation rate of flows, it therefore penalizes higher throughput applications more and assigns excessive rates to low data producing high priority flows. The maximum data generation rate at node *r4* is 13 kbps but the algorithm presented in [24] assigns it a rate of 18.161 kbps which shows the algorithms' erroneous behavior. Such errors degrade the performance of contending nodes flows, because they are deprived of the extra bandwidth that can be used by them.

The modified rates assigned to nodes *r4* and *r5* trigger the congestion control mechanism at node *r5*.

Table 13. Bandwidth share

Node ID	β_j
r4	18.161 kbps
r5	20.638 kbps

Table 14 shows the data rate assigned to different nodes by the congestion control algorithm presented in [24].

Table 14. Bandwidth assigned to nodes

β_a	β_b	β_c	β_d	β_e
11.21 kbps	6.406 kbps	6.990 kbps	6.214 kbps	6.214 kbps

Fig. 9 compares both algorithms in terms of rate adjustment at each source node. It shows that our proposed algorithm favours nodes producing the bulk of high priority data, as it does not penalize nodes *c* (because it gets the needed bandwidth and in [24] *c* gets more bandwidth than actually required), *d*, and *e*.

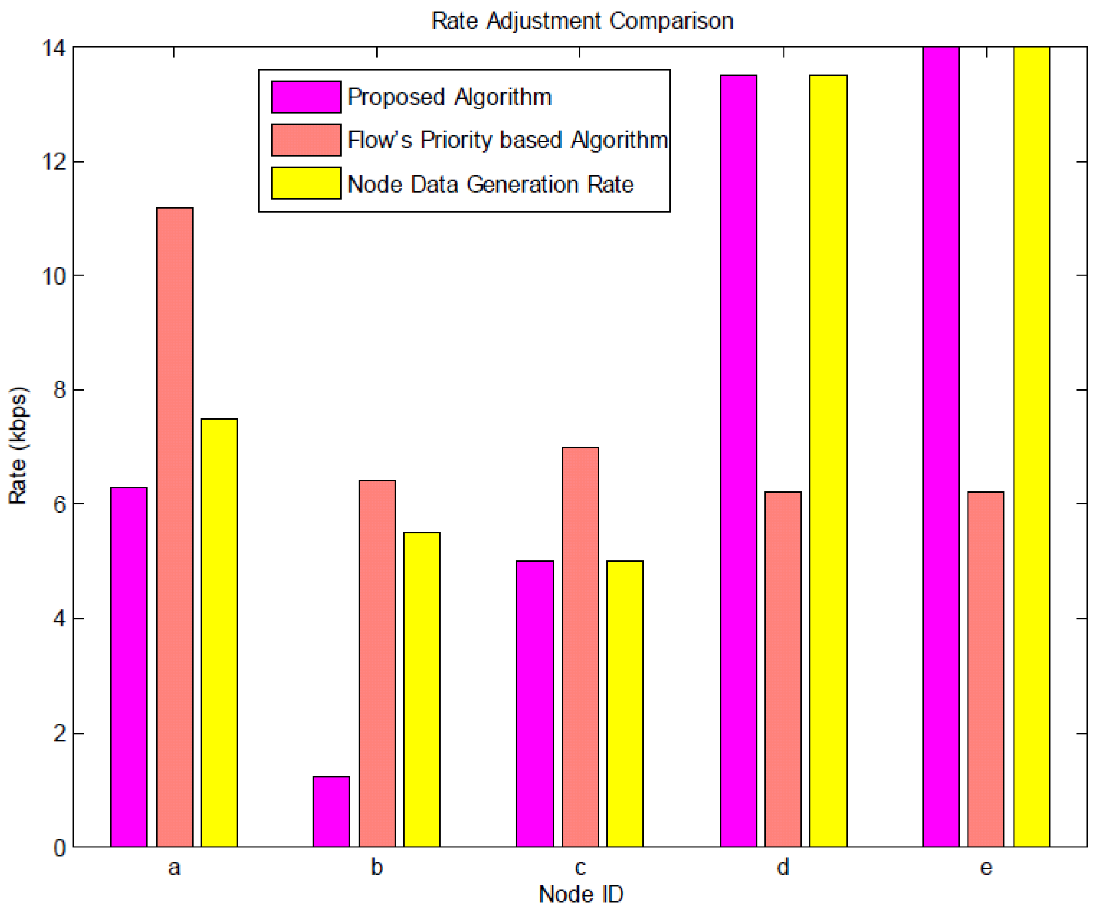

Fig. 9. Rate Adjustment Comparison

Conclusion

In this paper, we have presented a survey of the state-of-the-art in cross-layer architectures for WSNs along with the survey of cross-layer architectures for supporting QoS in WSNs and WMSNs. Afterwards, we introduced cross-layer architectures to support multiple types of applications in WMSNs. Then, in order to provide better QoS support in WMSNs, we proposed an admission control mechanism along with differentiated services support in WMSNs. Moreover, we have devised a differentiated services based congestion detection and control algorithm for WMSNs to support multimedia applications in case of congestion in the network. Analytical results show that our proposed algorithms can yield better perfomance, because during congestion, our algorithm calculates modified data rates not for itself but for all its upstream nodes for which it is acting as a relaying node. Moreover, while assigning the data rates, the proposed algorithm considers the prioties of data flows as well as their bandwidth usage.

References

[1] I. F. Akyildiz, W. Su, Y .Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: a Survey", *Computer Networks*, vol. 38, no. 4, pp. 393 - 422, 2002. Article (CrossRef Link).

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “A survey on wireless multimedia sensor networks,” *Computer Networks*, vol. 41, no. 4, pp. 921 - 960, March 2007. Article (CrossRef Link).

[3] A. Durresi, “Architectures for Heterogeneous Wireless Sensor Networks,” in *Proc. of IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1289-1296, 2005.

[4] K. Pentikousis, “TCP in Wired-cum-Wireless Environments,” *IEEE Communication Surveys and Tutorials*, vol. 3, no. 4, 2000. Article (CrossRef Link).

[5] P. Hurni, T. Braun, B. K. Bhargava, and Y. Zhang, “Multi-hop cross-layer design in wireless sensor networks: A case study,” in *Proc. of IEEE International Conference on Wireless and Mobile Computing, Networking and Communication*, pp. 291-296, 2008. Article (CrossRef Link).

[6] T. Melodia, M. C. Vuran, and D. Pompili, “The state of the art in cross-layer design for wireless sensor networks ,” *Springer Lecture Notes on Computer Science,* vol.3883, pp. 78-92, 2006.

[7] J. Z. Sun and J. Sauvola, "Cross-Layer optimization framework for wireless sensor networks,” in *Proc. of International Conference on Intelligent Robots and Systems*, pp. 2029-2034, 2006. Article (CrossRef Link).

[8] C. Lin, Y. X. He, C. Peng, and L. T. Yang, “A distributed efficient architecture for wireless sensor networks,” in *Proc. of 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW)*, pp. 429-434, 2007. Article (CrossRef Link).

[9] I. F. Akyildiz, M. C. Vuran, and O. B. Akan, “A Cross-layer protocol for wireless sensor networks,” in Proc. of *40th Annual Conference on Information Sciences and Systems*, pp. 1102-1107, 2006. Article (CrossRef Link).

[10] C. J. Merlin, C. H. Feng, and W. H. Heinzelman, “Information-sharing protocol architecture for sensor networks: the state of the art and a new solution,” in *Proc. of Mobile Computing and Communications Review*, vol. 13, no. 4, pp. 26-38, Oct. 2009. Article (CrossRef Link).

[11] D. Lin and S. Li, “TCLA: A triangular cross-layer architecture for wireless sensor networks,” in Proc. of *International Conference on Frontier of Computer Science and Technology*, pp. 272-278, 2009. Article (CrossRef Link).

[12] S. Alikhani, T. Kunz, M. St-Hilaire, and F. R. Yu, “A central-networked cross-layer design framework for wireless sensor networks,” in Proc. of *6th International Wireless Communication and Mobile Computing Conference*, pp. 301-305, 2010.

[13] L. V. Hoesel, T. Nieberg, J. Wu, and P. J.M. Havinga, “Prolonging the lifetime of wireless sensor networks by cross-layer interaction ,” in *Proc. of IEEE Wireless Communications*, vol. 11, no. 6, pp. 78-86, Dec. 2004. Article (CrossRef Link).

[14] D. G. Costa and L. A. Guedes, “A Survey on Multimedia-Based Cross-Layer Optimization in Visual Sensor Networks,” *Sensors,* vol. 11, pp. 5439-5468, 2011. Article (CrossRef Link).

[15] E. Troubleyn, E. D. Poorter, P. Ruckhebusch, I. Moerman and P. Demeester, “Supporting Protocol-independent adaptive QoS in wireless sensor networks,” in Proc. of *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pp. 253-260, 2010. Article (CrossRef Link).

[16] E. Troubleyn, E. D. Poorter, I. Moerman and P. Demeester, “AMoQoSA: Adaptive Modular QoS Architecture for wireless sensor networks,” in *Proc. of 2nd International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pp. 172-178, 2008. Article (CrossRef Link).

[17] M. H. Yaghmaee and D. Adjeorh, “A model for differentiated service support in wireless multimedia sensor networks,” in Proc. of *17th International Conference on Computer Communications and Networks*, pp. 1-6, 2008.

[18] M. Younis, K. Akkaya, M. Eltoweissy, and A. Wadaa, “On handling QoS traffic in wireless sensor networks," in *Proc. of 37th Hawaii International Conference on System Sciences*, pp. 1-10, 2004.Article (CrossRef Link).

[19] L. Shu, M. Hauswirth, Y. Zhang, J. Ma, G. Min, and Y. Wang, “Cross-layer optimization for data gathering in wireless multimedia sensor networks within expected network lifetime,”

Journal of Universal Computer Science, vol. 16 no. 10, pp. 1343-1367, 2010.

[20] H. S. Aghdasi, M. Abbaspour, M. E. Moghadam, and Y. Samei, "An energy-Efficient and high-quality video transmission architecture in wireless video-based sensor networks, " *Sensors,* vol. 8, pp. 4529-4559, 2008. Article (CrossRef Link).

[21] S. Meguerdichian , F. Koushanfar, M.Potkonjak, and M. B. Srivastava, "Coverage problem in wireless ad hoc sensor networks," in *Proc. of IEEE INFOCOM*, pp. 1380 - 1387, 2001.

[22] Tora Linear Programming: http://www.dl4all.com/jug/Tora+Linear+Programming.html.

[23] M.O. Farooq and A.S. Butt, "Hybrid differentiated service architecture for qoS routing in mobile ad hoc networks," in *Proc. of the IEEE 13th International Multitopic Conference* (INMIC), pp. 1-6, pp. 14-16 Dec. 2009.

[24] M. H. Yaghmaee and D. A. Adjeroh "Priority based rate control for service differentiation and congestion control in wireless multimedia sensor networks", *Computer Networks*, vol. 52, no. 11: pp. 1798-1811, 2009. Article (CrossRef Link).

Muhammad Omer Farooq has received his BS (Computer Science) and MS (Computer Engineering) degrees with distinction from the Virtual University of Pakistan and the Center for Advance Studies in Engineering (CASE) Islamabad, Pakistan respectively. He is a PhD student in the Institute of Telematics, University of Lübeck, Germany. His research focuses on algorithms and protocols for wired and wireless computer networks. In recent years, his research has focused on QoS provisioning in the Internet, Mobile Ad-hoc Networks, and Wireless Multimedia Sensor Networks.

Marc St-Hilaire is currently an Associate Professor in the School of Information Technology with a cross appointment to the Department of Systems and Computer Engineering at Carleton University. He received the Computer Engineering degree from Concordia University in 2002 and the master and Ph.D. degrees in Computer Engineering respectively in 2003 and 2006 from the École Polytechnique de Montréal. His research interests include telecommunication network planning, mathematical modeling, performance analysis, and mobility management.

Thomas Kunz is currently a professor in the department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. He heads the Mobile Computing Group, researching wireless network architectures (manets, wireless mesh networks, and wireless sensor networks), network protocols (routing, mobile IP, and QoS support), and middleware layer for innovative wireless applications. He has served on more than 50 TPCs of international conferences and workshops in the mobile and wireless domain. He is author or coauthor of more than 160 technical papers and received number of awards and best paper prizes. He is sensior member of both IEEE and ACM.