

# Chapter 12

## Quality of Service in Wireless Sensor Networks

Can Basaran and Kyoung-Don Kang

**Abstract** Although well studied for traditional computer networks, quality of service (QoS) concepts have not been applied to wireless sensor networks (WSNs) until recently. QoS support is challenging due to severe energy and computational resource constraints of wireless sensors. Moreover, certain service properties such as the delay, reliability, network lifetime, and quality of data may conflict by nature. Multi-path routing, for example, can improve the reliability; however, it can increase the energy consumption and delay due to duplicate transmissions. Also, high resolution sensor readings incur more energy consumptions and delays. Modeling such relationships, measuring the provided quality, and providing means to control the balance is essential for QoS support. In this context, this chapter discusses existing approaches for QoS support in WSNs and suggests directions for further research.

### 12.1 Introduction

Quality of service management refers to systematic approaches to measuring and managing the quality of computational services. It has recently attracted a lot of interest, especially producing abundant research results in wired networks. A QoS study investigates the interplay between various service parameters such as bandwidth allocation and their impact on the provided service quality such as delay, jitter, and/or throughput. Reservation-based approaches such as IntServ [1] and reservation-less approaches such as DiffServ [2] are developed in order to provide bandwidth guarantees and service differentiation, respectively. IntServ architecture specifies a flow-based bandwidth reservation protocol to provide seamless data

---

C. Basaran (✉)

Department of Computer Science, Thomas J. Watson School of Engineering and Applied Science, State University of New York at Binghamton, P.O. Box 6000, Binghamton, NY 13902-6000, USA  
e-mail: cbasaran@cs.binghamton.edu

streams to users. On the other hand, DiffServ architecture does not maintain per-flow status. Instead, it supports differentiation between service classes to provide better QoS, e.g., shorter delay and smaller jitter, for a service class with a higher priority.

Mobile Ad hoc NETWORKS (MANETs) have different challenges for QoS support [3]. The differences stem from dynamic topologies, relatively low bandwidth, and shared wireless communication medium associated with MANETs. Despite all these differences, most QoS studies in MANETs focus on bandwidth allocation [4]. The fact that MANET topologies are more stable and nodes in such topologies are more capable than those of WSNs distinctly differentiates between the two network domains.

WSNs have an entirely different architecture. Individual nodes in a WSN have severe resource constraints in terms of energy, network bandwidth, memory, and CPU cycles. Also, they have unstable radio ranges, transient connectivity, and unidirectional links [5]. Despite these constraints, WSNs are often deployed for mission critical applications. These properties highlight the importance of QoS in WSNs [6]. Unfortunately, existing QoS approaches are not directly applicable to WSNs. For example, flow-based approaches such as IntServ need to establish end-to-end connections; however, an individual sensor node does not have sufficient resources to manage the state information per connection. Moreover, unstable connectivity between nodes makes it impossible to establish a persistent path between two distant ends.

A WSN acts as a collective unit to provide a sensor data service such as target tracking, fire detection, or habitat monitoring. Notably, QoS requirements, e.g., the required accuracy of sensor readings and the importance of a single reading, vary greatly from application to application. For instance, a routing protocol in a short-term target tracking application can be tuned to minimize the delay and maximize the reliability via real-time multipath routing for increased energy consumptions. For fire detection in a smart building, reliability is critical to ensure that important sensor readings are not lost. However, timeliness can be differentiated based on data values such that high temperature or pressure readings receive higher priority than normal readings. Also, redundant data indicating normal status can be aggressively aggregated to minimize energy consumptions. Further, a WSN deployed for long-term habitat monitoring may not need to support real-time data transmission. Data can be aggregated to minimize the energy consumption and stored at the base station to be sent to scientists every day, for example, via a satellite connection. Therefore, establishing a QoS model based on a specific application scenario allows us to identify key QoS requirements and metrics from which a feasible QoS management scheme potentially involving trade-offs can be derived. At the same time, it is important to identify key QoS requirements, if any, which apply to most WSN applications. Overall, QoS support in WSNs is a fairly new research problem with many remaining issues to investigate. In this section, we give a survey of well-known existing work, while discussing QoS issues in WSNs for future work.

## 12.2 Background

The quality of information provided by a WSN, e.g., accuracy, timeliness, or reliability, and the overall lifetime of the WSN are two major, conflicting properties. It has been reported that each link on a path increases the average packet loss ratio by approximately 5–10% depending on the node density and communication pattern [7], which results in the loss of approximately half of the packets along a path of 15 nodes. The same study [7] identifies that the shortest round-trip-time is approximately 600 ms, while the largest delay is approximately 5 s for a WSN covering an area of 1,200 m<sup>2</sup>. This is a clear indication of unstable behaviors of wireless communication media. Guaranteeing QoS in such a medium is certainly a challenging issue [8].

It is possible to craft tailored solutions for specific application needs and avoid generic approaches. But even so, an application often needs to execute in different states, while having flows with different priorities including control messages, periodic sensor readings, and alert messages. To provide network-wide QoS, each system component must comply with the desired QoS parameters. In this chapter, MAC (medium access control) layer, network layer, and in-network processing solutions for QoS support in WSNs are discussed. A number of these approaches are cross-layer solutions, since it is not always possible to divide system components into mutually exclusive modules. In fact, the reflections of the open systems interconnect (OSI) layers have a tendency to melt into each other in WSNs [9].

### 12.2.1 MAC Layer Solutions

The MAC layer provides channel access control services, which allow nodes to share the multiaccess wireless communication channel. Most network layer QoS solutions in the timeliness domain have MAC layer extensions. These extensions include, but are not limited to, modifications of the CSMA/CA protocol such that the back-off delay is inversely proportional to the priority of the packet being sent. Hence, upon a collision, a node with a high-priority packet to transmit waits for a shorter time interval before retrying to gain access to the wireless channel.

Since retransmissions are handled in the MAC layer in case of a transmission failure, upper layers may need to query the MAC layer in order to obtain information on the congestion state and the link quality. Most routing solutions examined in Sect. 12.2.2 utilize this information for delay estimation. Overall, MAC level QoS support is mostly limited to the policies implementing scheduling, channel allocation, buffer management, error control, and error recovery. The MAC layer QoS support in WSNs particularly focuses on scheduling and channel allocation to support upper layer services, such as routing and data aggregation, as discussed next.

*QUIRE* [10] is a cluster-based MAC protocol trying to form node clusters such that only one node in a cluster sends sensed data by communicating with a mobile agent flying over the deployment area. This approach focuses on WSNs with mobile

agents that partition the region into hexagonal cells. Each agent broadcasts the cell radius, hovering over each cell. Each node in the cell receives this message and waits for a period inversely proportional to the quality of the received message. During the wait period, if a node hears a reply to the broadcast message from a node in the same cluster but with a better link quality, it cancels its own transmission. This approach aims at collecting sufficient data from the network such that the data distribution in the sensed field can be regenerated with a given probability  $P_s$ . At the same time, QUIRE ensures that the point of sensing can be estimated with a mean square error less than the *maximum distortion*  $D$ . Cell partitioning is accomplished taking into account the QoS metrics  $P_s$  and  $D$ .

*Q-MAC* [11] is an energy-efficient, QoS-aware MAC protocol for WSNs designed to offer service differentiation between two classes via intra- and inter-node scheduling. Each node has an intra-node classifier which uses a separate FIFO queue for each priority level. Inter-node level classification gives the channel to the node with the highest urgency. The urgency of a node is evaluated by considering its packet priority, remaining number of hops to the destination, remaining energy of the node, and the queue lengths.

*CC-MAC* [12] exploits *spatial correlation* of sensor readings by pruning redundant data. Since sensor nodes within certain proximity generate similar data, based on the statistical information on node distribution, a *correlation radius* is calculated. This radius is used by CC-MAC to define *correlation regions* and perform filtering of messages belonging to the same correlation region. CC-MAC protocol is composed of two major components: Event MAC (E-MAC) and Network MAC (N-MAC). While E-MAC reduces the in-network traffic by dropping packets in the same proximity, N-MAC deals with forwarding filtered packets to the sink and prioritizes packets coming from foreign proximities. Although left as a future work, the QoS implications of proposed approach are evident, as the *correlation radius* may be tuned according to user-defined accuracy constraints.

An *Implicit Prioritized Access Protocol for WSNs* [13] defines a MAC layer protocol for cell-structured networks. The protocol provides delay guarantees for message delivery, while fully utilizing the available bandwidth via an earliest deadline first (EDF) scheduler, which exploits the periodic nature of WSN messages. In this approach, nodes are grouped into cells. All nodes within a cell are directly linked with each other. Intercell communication is handled by more capable *cluster heads* (CHs). A CH has two transceivers to transmit and receive packets at the same time. A total of seven radio channels are used in the whole network, which is modeled as a collection of hexagonal cells. A CH can communicate with the sensors in its cell, while communicating with the CHs of the (maximum six) neighboring cells using different channels. Intracell communication is based on a shared EDF schedule. The shared nature of this schedule allows all nodes to know precisely who should talk, and when. Moreover, some time slots are reserved for intercell communication. Using these time slots, CHs talk with each other based on another EDF schedule. If a node will not use the rest of the slots allocated to it, it broadcasts a yield message for the remaining slots. In this case, the next eligible node can take over the channel, increasing the bandwidth utilization.

### 12.2.2 Network Layer Solutions

MAC layer protocols can handle one-hop communication. For end-to-end QoS guarantees, network layer support is needed. The network layer QoS in WSNs encompasses end-to-end real-time service and reliability, which are fundamental requirements for mission-critical WSN applications. Because of energy demanding nature of radio transmissions [14, 15], QoS-aware routing protocols also have to use minimum number of control messages. It is challenging to support the desired QoS, while minimizing the number of control messages.

Given the scarce resources of sensor nodes [16], implementing an efficient routing protocol with no help from the underlying MAC layer is a daunting task. It follows that most of the approaches in this category are *cross-layer* solutions, which are built upon a QoS-aware MAC protocol to leverage lower level network information and service needed for QoS provision at the network layer.

In the reliability domain, using multipath routing [17–19] is a common approach. The idea behind this scheme is to exploit the high node density prevalent in WSNs. Because of the high density, there could be multiple paths between the source and the sink. If we assume that the packet delivery ratio for a link is 95%, then the delivery ratio for a 14-hop path is less than 50%. However, if there is a second, disjoint path with the same number of hops and link reliability, packets can be duplicated along the two paths to achieve a delivery ratio of 75%. The increase in reliability is achieved by sacrificing network lifetime, since energy consumption is roughly doubled if two paths rather than one are used. Alternatively, multipath routing can support load balancing to increase the network lifetime [20]. To transmit a packet, in this case, the routing algorithm only selects a single path among multiple paths for load balancing.

*RAP* [21] proposes a cross layer, i.e., network and MAC layers, architecture designed to support soft real-time requirements in WSNs. The architecture employs geographic forwarding (GF) [22, 23], in which a node forwards a packet to its one hop neighbor that is closer to the sink than the node is. When there are multiple one hop neighbors closer to the sink, it forwards the packet to the node that is closest to the sink. Thus, a node only has to keep the geographic locations of its one-hop neighbors.

In *RAP*, the sensing period and deadline are specified for each query. *RAP* applies Velocity Monotonic Scheduling, in which the priority of a data packet responding to a query is determined according to its requested  $velocity = distance/deadline$ . Specifically, in Static Velocity Monotonic (SVM), a permanent priority is assigned to a packet at the source according to the required velocity. On the other hand, Dynamic Velocity Monotonic (DVM) supports dynamic velocity adjustment at relay nodes. In DVM, when an intermediate node receives a packet, it assigns a new priority to the packet according to the remaining distance to the sink and time to the deadline. Therefore, when a packet suffers congestion, its priority, i.e., velocity, can be increased. On the contrary, if a packet moves faster than the requested velocity, its priority is decreased providing more bandwidth to others. This approach, however, requires either time synchronization [24] or MAC layer

support for elapsed time calculation. In [21], DVM underperforms SVM possibly due to the lack of a reliable mechanism to measure the in-network delay and adjust the velocity accordingly. RAP requires prioritized MAC, which differentiates back-off delays according to the packet priority. Upon a collision, a node picks a random back-off delay in the interval  $[0, CW)$ . The contention window  $CW = CW_{\text{prev}} \times (2 + (\text{PRIORITY} - 1)/\text{MAX\_PRIORITY})$  where  $CW_{\text{prev}}$  is the previous contention window size and  $\text{MAX\_PRIORITY}$  is the number of priority levels similar to [25]. As a result, a high-priority packet is likely to have a shorter back-off interval upon a collision.

Greedy GF is not always possible in the presence of a void, in which neither of the one hop neighbors is closer to the sink than the node that currently has the packet [22,23]. However, RAP simply assumes the constant availability of GF and it lacks void avoidance logic. It also has no congestion control mechanism. As a result, many packets with high-velocity requirements may miss their deadlines when the network is congested. Furthermore, the study does not consider the dynamic nature of WSN links and nodes in detail.

*SPEED* [26] is a routing protocol that aims to provide a uniform delivery speed across a WSN. Similar to RAP, *SPEED* relies on GF. Unlike RAP, it does not depend on MAC level real-time support. Each node keeps the location information of its one-hop neighbors and delay estimation for each link, which is computed using regular data messages and corresponding delays piggybacked on the acknowledgments (ACKs).

The QoS parameter *delivery speed (SetSpeed)* is supported in a best effort manner. Each node forwards a packet to a one-hop neighbor that is closer to the sink and to which it is connected via a wireless link that supports the *SetSpeed*. A neighbor supporting a higher speed is more likely to be selected. This forwarding approach is called the *stateless nondeterministic geographic forwarding*. If the required speed cannot be supported, packets are dropped with a given probability called *relay ratio*, which is calculated by the *neighborhood feedback loop* at the MAC layer based on the measured packet loss information indicating the severity of congestion or bad link quality.

When a node has no forwarding candidate, which is closer to the sink than itself or it cannot satisfy the desired speed to a specific destination, the node performs *backpressure* routing. The node issues a backpressure beacon to upstream nodes to notify them of the average delay suffered by the link on the path to the sink. Nodes receiving this information update their tables with the new information. If a node does not have the issuing node as a candidate to the destination it ignores this backpressure beacon. Voids are also avoided on the fly using backpressure beacons. A node identifying a void sets the average delay to infinity and informs upstream nodes.

*Multipath Multi-SPEED* (MMSPEED) [17] is a cross-layer protocol encompassing the network and MAC layers. It extends *SPEED* by providing *multiple network-wide speed levels for service differentiation*, while supporting QoS in the *reliability* domain at the same time. For scalability, MMSPEED relies on GF, similar to RAP and *SPEED*. The key idea is to have different speed layers over a single

network. Hence, for  $N$  speed layers, there are  $N$  different *SetSpeeds*. Each virtual layer has its own FCFS queue. Different from SPEED, the MAC layer in MMSPEED prioritizes packets belonging to higher speed layers over those of lower speed layers. Additionally, each node computes the remaining time to a packet's *deadline* and dynamically sets a new speed layer for the packet such that the new speed is the minimum speed able to satisfy the deadline requirement.

In the *reliability* domain, MMSPEED leverages the *loss rate* information provided by the MAC layer and *multipath routing*. Assuming a homogeneous loss rate across the network and a homogeneous hop distance, node  $i$  locally estimates the end-to-end reachability (RP) of a packet to destination  $d$  through one-hop neighbor node  $j$  as follows:

$$RP_{i,j}^d = (1 - e_{i,j})(1 - e_{i,j})^{[\text{estimated number of hops}]}, \quad (12.1)$$

where  $e_{i,j}$  is the known one-hop loss rate for the link between nodes  $i$  and  $j$ . The hop count in (12.1) is estimated by dividing the known distance to the final destination by the known one-hop distance to node  $j$ . Thus, the last part in (12.1), i.e.,  $(1 - e_{i,j})^{[\text{estimated number of hops}]}$ , is a rough estimate for the rest of the network. Given a required reliability  $P^{\text{req}}$ , node  $i$  can forward the packet to node  $j$  if  $RP_{i,j}^d \geq P^{\text{req}}$ . Since decisions are based on estimates that can later prove to be incorrect, dynamic compensation logic is also implemented. When a node  $s$  cannot find a single neighbor satisfying  $P^{\text{req}}$ , it can choose to forward the packet to two nodes ( $j_1$  and  $j_2$ ). In this example, if  $P^{\text{req}} = 80\%$ ,  $RP_{s,j_1}^d = 70\%$ , and  $RP_{s,j_2}^d = 60\%$ , then node  $s$  will calculate total reaching probability (TRP) as follows:

$$\begin{aligned} \text{TRP} &= 1 - (1 - RP_{s,j_1}^d)(1 - RP_{s,j_2}^d), \\ &= 1 - (1 - 0.7)(1 - 0.6) = 0.88. \end{aligned}$$

where  $(1 - RP_{s,j_1}^d)$  is the probability that the path through node  $j_1$  will fail and  $(1 - RP_{s,j_2}^d)$  is the probability that the path through  $j_2$  will fail. Thus, TRP is the probability that at least one of them will deliver the packet toward the sink. Node  $s$  can arbitrarily assign a new  $P^{\text{req}}$  (e.g., 0.6 and 0.5) to each node because  $\text{TRP} = 1 - (1 - 0.6)(1 - 0.5) = 0.8$ . Similar to the case of the *timeliness domain*, when a node suffers the existence of unreliable neighbors, it can use *reliability backpressure beacons* to decrease the expectations of upstream nodes. In this case, the node issuing a backpressure beacon will not be assigned a reliability level more than that specified in the beacon message. As the effect of a backpressure beacon only lasts for a limited time period, the impact of transient link problems on reliability assessment is limited.

By serving each level with just enough speed and reliability, MMSPEED efficiently utilizes precious resources.

*JiTS* (Just-in-Time Scheduling) [27] is a network layer protocol for soft real-time packet delivery. JiTS only considers *timeliness* without considering reliability. It does not assume the underlying support of a QoS-aware MAC. Instead, it relies on the widely accepted nonprioritized IEEE 802.11 MAC. Unlike other routing protocols, JiTS aims at delaying a packet as much as its deadline allows. The

idea of delaying packets is similar to the just-in-time delivery concept proposed in Mobicast [28] designed for mobile users of sensor data. Unlike Mobicast, JiTS does not assume user mobility. Also, it can be useful especially when in-network data aggregation is employed. Exploiting the slack-time increases the probability of similar data meeting at a relay node for aggregation.

JiTS forwarding logic uses a sorted queue where packets are inserted in nondecreasing order of target transmission time. The packet at the head of the queue is forwarded when the transmission time is reached. Target transmission time is calculated using the average one-hop delay estimated via ACK messages and estimated number of hops to the destination. The slack time determining the target transmission time (= current time + slack time) is estimated and uniformly distributed over all hops in the path:

$$\text{Slack time} = \frac{(\text{Deadline} - \text{EETD})}{\text{distance}(X, \text{sink})} \times a, \quad (12.2)$$

where EETD is the estimated end-to-end transmission delay, which is equal to the product of the estimated average one-hop delay and the estimated number of hops to the destination. Also, variable  $a$  in (12.2) is the *safety factor*. By setting it to a value less than 1 such as 0.7, JiTS can tolerate estimation errors.

JiTS has several variations. Especially, *nonlinear JiTS* shows the best performance among them. In a WSN, congestion may increase as packets approach the sink due to many-to-one communication patterns from sources to the sink. To reduce potential contention near the sink, nonlinear JiTS delays a packet more as the packet gets closer to the sink. Specifically, exponentially increasing portions of the estimated slack are allocated to the nodes closer to the sink:

$$\text{Slack time} = \frac{(\text{Deadline} - \text{EETD})}{2^{R/O}} \times a, \quad (12.3)$$

where  $R$  is the remaining distance to the sink and  $O$  is the estimated one-hop distance.

Unlike RAP, SPEED, and MMSPEED, JiTS does not assume any specific routing protocol. In their simulation study, the shortest path routing protocol supported by many WSN systems such as TinyOS considerably outperforms GF in terms of deadline miss ratio and packet drop ratio.

*LESOP* (Low-Energy Self-Organizing Protocol) [9] is built based on a new two-layered network architecture called embedded wireless interconnect (EWI) replacing the OSI model. The design of EWI is justified by the fact that almost all solutions in WSNs require cross-layer implementations.

LESOP was specifically designed for target tracking applications, in which the first node detecting the target initiates the cooperation among nodes by broadcasting busy tones through a secondary wake-up radio channel. LESOP focuses on the accuracy of the target location by modeling trade-offs between QoS, i.e., the accuracy of the target location, and energy consumption. Increasing the idle time between sensing intervals decreases energy consumption, but it increases the delay for target



detection. This is an evident trade-off between energy and QoS. Moreover, LESOP models the relationship between the target tracking error and the coverage to determine the minimum number of nodes that should be in sensing state. This minimum is calculated by using a QoS knob that is the minimum acceptable gain achieved by adding a new node to the sensing set. A drawback of LESOP is requiring a secondary radio, which can increase the cost.

### 12.2.3 In-Network Data Services

Since sensor readings are often redundant, sensor data can be aggregated in the network to reduce the number of packet transmissions and corresponding energy consumptions [29]. Such services can be implemented as a part of a user level application or a separate data service layer [30]. This domain deals with the quality and accuracy of the sensed data as well as minimization of in-network traffic, while conforming to a predetermined sensing accuracy. It is important to take into consideration that the process of aggregation may violate real-time constraints [48], because it requires relaying nodes to delay messages in order to aggregate them with data from different nodes. Thus, data aggregation should cooperate with QoS-aware routing and MAC to maximize the effect.

*Prediction-based monitoring in sensor networks* (PREMON) [31] applies the principles of MPEG [32] compression to the field of WSNs. In this approach, the sink node accumulates sufficient information to construct a *prediction model*. It then distributes this model to appropriate sensors along with the lifespan associated with this model. Sensor nodes receiving the prediction model change the mode of sensing to *update mode* and begin to send their sensor readings only when they differ from the predicted values by more than a predefined error margin. This is how a requested quality of monitoring (QoM) is provided.

PREMON is one of the first approaches to predicting sensor readings for the reduction of radio transmissions. The reduction depends on the *error tolerance*, i.e., the predefined QoM, and the correctness of the prediction model. Frequent distribution of a prediction model by a base station may consume significant amount of precious energy. Also, the scalability of PREMON is limited due to the centralized model construction.

*Temporal Coherency-Aware in-Network Aggregation* (TiNA) [33] proposes an approach to exploiting the *temporally coherent* nature of sensor readings. Each query has a *tct* (temporal coherency tolerance) value specified within the query itself. If the difference between the new reading and old one is smaller than the associated *tct* value, sensors do not report their readings. Parent nodes keep track of their child nodes while trying to aggregate their readings. If the parent does not receive an update from any of its children, the old reading from the child (or children) is used for aggregation. To distinguish a failed node from a node remaining silent because of the TiNA logic, a node sends periodic *heartbeat* messages to the parent. In this way, TiNAS's approach increases the quality of data (QoD) when network suffers severe congestion.

Romer et al. [34] reduce the amount of data to transmit based on the QoS requirements. A source and sink pair defines the tolerable error budget  $e_{\max}$ . Rather than transmitting a complete stream of sensor data  $\{x[k]\}$  from the source to sink, the source only sends a subset of data to the sink. More specifically, both the source and the sink run the same least-mean-square (LMS) predictor [35]. Using the LMS predictor, the source node computes the prediction error  $e[k] = x'[k] - x[k]$ , where  $x'[k]$  is the sensor data predicted by the LMS method and  $x[k]$  is the actual sensor reading. The source transmits  $x[k]$  only if  $e[k] > e_{\max}$ . Otherwise,  $x[k]$  is simply dropped. Thus, a pair of the source and sink can support  $e_{\max}$ .

### 12.3 Thoughts for Practitioners

WSNs with limited resources are often deployed for mission-critical applications. Thus, QoS-aware approaches are investigated to improve the cost–benefit ratio. Unfortunately, providing the desired QoS is not as straightforward as implementing a service. Services such as routing, MAC layer protocols, localization, time synchronization, and in-network data aggregation can be implemented in isolation, neglecting other system parameters. In the quality domain, all services share at least a common subset of interest such as network lifetime and delay. Essentially, the notion of *quality* denotes a black-box model, in which the end user expects a seamless integration of services that can be expressed in terms of inputs and outputs.

Is QoS a byproduct of a protocol, a result of a final skimming over a proposed solution, or the ultimate objective at the design phase? It is important to answer this question because, in the two former cases, QoS is only a minor issue in the application of interest. If QoS is the major concern, the integration and cooperation between the components and the resulting overall system performance matter. For example, a routing solution aiming to support soft real-time delivery guarantees may perform well in simulations; however, it might be useless in a real system, because it ignores global parameters of a real WSN system such as the MAC protocol and data aggregation, which can considerably affect the service delay.

Generally, QoS is a system-wide concept and it has to be handled as such. Because of severe resource constraints, most WSN protocols are optimized for specific applications of interest. Further, most solutions are cross-layered. As a result, QoS management becomes complex. Hence, a promising approach for QoS support is to analyze the application domain and design a QoS solution that conforms with the basic approaches employed by the intended set of target applications. A summary of key issues related to QoS-aware service design follows:

- QoS support must be an integrated part of the whole design/development process. Thus, a detailed analysis of the intended set of applications must be the starting point. The design should be compatible and cooperative with possible domain-specific system configurations unless it is generally applicable to different application domains.

- The QoS parameters to be supported must be decided considering their relevance to the target set of applications. Also the performance metrics and measurement methods, tools, and environments should be determined. If a cross-layer approach is taken, the required lower-layer services offered to the upper layers must be considered.
- Next, factors affecting the service performance for the selected QoS parameters must be investigated. For example, available wireless bandwidth and residual energy may affect data timeliness and reliability. In this way, the designers(s) of a QoS-aware WSN service can identify potential trade-offs between QoS parameters such as timeliness and reliability while considering a QoS model using the QoS parameters.
- Resource constraints such as memory and energy limitations should be considered. Because of severe resource constraints, a simple, lightweight approach is preferred. Also, one should take into account the nondeterministic, unreliable nature of wireless communication to let the QoS management scheme adapt to varying environments.
- Verifying models through simulations is important. If simulation results are convincing, before real-world deployment, the approach should be evaluated in a test bed composed of real battery-powered, wireless sensors such as MICA motes. Although there are many highly reliable simulation environments such as ns-2, virtually none of them can provide all the insights that can be gained from a testbed. On the other hand, a simulation study can cover a larger scope of experimental parameter settings and perform potentially intrusive or destructive experiments. Hence, simulation and testbed experiments are complementary to each other. From these experiments, issues that are important but were overlooked at design time can be newly identified to further improve the system design.
- If the previous steps are successful, the system can be deployed in a target environment starting from a relatively small-scale environment moving to a larger scale environment in a stepwise manner. The previous design steps may have to be revisited if new issues arise.

## 12.4 Directions for Future Research

Integration of various QoS functionalities within a system is an open research topic. We believe that future research efforts will follow this path and adopt a *holistic view* of QoS in WSNs. Definition of a holistic approach does not fall under the umbrella of a specific service category such as routing or channel allocation. Instead, it is a broader concept. A problem arises from the lack of established set of protocols even for specific application domains. Although there are groups trying to integrate existing research efforts [36] and proposals of complete systems [37], more work on service integration is required. In terms of QoS, neglecting *seemingly irrelevant*

system functionalities is not a recommended approach, because they could affect each other. Thus, QoS-oriented approaches need to go beyond microscopic views and will embrace cross-cutting issues for seamless integration.

An appropriate place to start QoS integration is the operating system, which needs to provide necessary interfaces enabling access to information, such as link quality and delay, required by QoS management schemes at upper levels. If the QoS-centric operating system provides necessary information as well as low-level services for QoS management, QoS solutions in the upper layers can focus on certain aspects directly related to them. For these reasons, a QoS-centric operating system for WSNs needs to define rich and clean interfaces between system services. Such an operating system should also be modular so that only the necessary system components can be selectively integrated for a specific application. By being composable and providing basic low-level system information and services widely used for QoS management, QoS-centric operating system can provide a basis for a holistic QoS management in WSNs.

Furthermore, new languages or extensions over existing languages [38–42] are necessary to address WSN-specific challenges. For example, it is essential to compose event-driven services in WSNs. Compared with traditional programming languages, WSN programming languages, e.g., nesC, are relatively hard to understand and program. A new language is needed to directly support the event-driven nature of WSNs, while reducing the difficulty of programming.

Multimedia-based sensing in WSNs is also important [43, 44]. Multimedia data, in forms of snapshots, audio, and video require strict QoS support from the network. New QoS-compatible media formats [45], new collaborative distributed in-network data processing algorithms [46], and new real-time services [47] are required to support demanding multimedia services. In a near future, multimedia sensing may become one of the major research topics in WSNs. Currently the related work is scarce.

## 12.5 Conclusions

This chapter discusses state-of-the-art approaches for QoS support in WSNs. A number of existing approaches for MAC, routing, and data services are investigated. Most research effort in this specific field is devoted to routing services for timeliness. As WSN research is relatively new, QoS issues are not fully investigated in WSNs. Key services such as MAC, routing, and data aggregation can be further extended to support QoS. Moreover, seamless integration of available approaches for WSN QoS management at different layers is not studied in depth. A holistic view is required to thoroughly investigate QoS interactions between different layers. If these approaches are integrated without enough care, they can adversely affect each other causing undesirable results. For example, excessive data aggregation can significantly reduce the timeliness, while real-time multipath routing may consume too much energy when applied inappropriately. Thus, care should be taken

to consider relevant QoS parameters in the context of an application of interest. At the same time, more research efforts are required to develop a general QoS model. QoS management mechanisms of the model could be composed to meet the needs of a specific application. In the future, bandwidth-demanding application scenarios such as multimedia sensing may further complicate QoS requirements in WSNs. This is another reason that a holistic approach is required for QoS management in WSNs.

**Acknowledgment** This work was partly supported by a NSF grant CNS-0614771.

## Terminologies

*Quality of service.* QoS may employ many different meanings such as delay, jitter, throughput, reliability, sensor data accuracy, or network lifetime. A QoS study defines appropriate QoS metrics for an application of interest and develops approaches to supporting the desired QoS via trade-offs. It is also desired for QoS research to identify and develop common QoS metrics and QoS management schemes for a broad range of applications. In summary, QoS management is a systematic approach to measuring and managing the quality of computational services.

*Flow-based QoS.* Flow-based QoS supports the desired QoS such as the required bandwidth for an established end-to-end connection. It can support deterministic QoS guarantees by maintaining per-flow state information and reserving resources needed for QoS support. However, it has poor scalability due to large overheads for managing per-flow information. IntServ is a representative protocol providing flow-based QoS.

*Class-based QoS.* Class-based QoS is developed to address the scalability problem of flow-based QoS. Instead of supporting per-flow QoS, it provides QoS for aggregate traffic classes. DiffServ [2] is a representative example.

*Hard-QoS.* Service is subject to strict and deterministic quality guarantees. Resources are reserved for service guarantees. The system will reject requests that cannot be satisfied due to a resource shortage.

*Soft-QoS.* No hard guarantee but a probabilistic guarantee of QoS is provided. Because of unreliable wireless communication and severe resource constraints, it is infeasible to support hard QoS guarantees in WSNs. Rather, soft statistical QoS guarantees are needed in WSNs.

*Timeliness.* Timeliness measures the degree of timely delivery of data. This is a critical QoS metric in a number of WSN applications, requiring real-time sensing and control.

*Reliability.* Reliability measures the delivery ratio of requested data. This QoS metric is also very important to ensure the reliable delivery of sensor data to the sink at which more sophisticated data analysis can be performed.

*Multipath routing.* Multipath routing refers to a class of routing protocols utilizing more than one path for data communication. For reliability, a single data packet can be transmitted through multiple paths. Alternatively, for load balancing, one packet is forwarded through a single link at a time even if there are multiple links available.

*Quality of data (QoD).* QoD refers to the quality of information provided such as the data accuracy, resolution, and timeliness. Since WSNs are data centric, QoD is a broader concept than the traditional notion of QoS, which mainly focuses on the low level network performance such as the delay, jitter, or throughput.

*Temporal coherency.* Sensor data values do not largely oscillate within a given time interval. For example, temperature readings in a smart building may not change drastically from one sensing period to another. This property can be leveraged for data aggregation.

## Questions

1. Do you think reservation-based QoS provision is applicable to WSNs? Why or why not?
2. Give two specific application scenarios with different QoS expectations.
3. What are the main differences between wired networks, infrastructure-based wireless networks, mobile ad-hoc networks (MANets), and WSNs in terms of QoS?
4. What is the main focus of QoS efforts in the MAC layer?
5. What is the main focus of QoS efforts in the network layer?
6. What is data aggregation? How can it be utilized as a QoS tool?
7. How can temporal coherency of sensor readings be exploited to satisfy different QoD demands?
8. How can spatial coherency of sensor readings be exploited to satisfy different QoD demands?
9. What is the main difference between timeliness and reliability domains when served over multiple paths?
10. What is the most widely used approach to calculating path delay in routing? Why?

## References

1. R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: An Overview. IETF RFC 1633, 1994
2. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. IETF RFC 2475, 1998
3. P. Mahapatra, J. Li, and C. Gui. QoS in mobile ad hoc networks, IEEE Wireless Communication, vol. 10, no. 3, pp. 44–52, 2003

4. K. Wu and J. Harm, QoS support in mobile ad hoc networks, *Crossing Boundaries*, vol. 1, no. 1, Fall 2001
5. I.F. Akyildiz et al. *Wireless sensor networks: A survey*, *Computer Networks*, Elsevier Science, vol. 38, no. 4, pp. 393–422, 2002
6. Y. Wang, X. Liu, and J. Yin. Requirements of Quality of Service in Wireless Sensor Network. *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICON-SMCL'06)*, Mauritius, 2006
7. N. Ota, D. Hooks, P. Wright, D. Auslander, and T. Peffer. Poster Abstract: Wireless Sensor Network Characterization – Application to Demand Response Energy Pricing, In *Proceedings of the First international conference on Embedded Networked Sensor Systems*, November 05–07, 2003
8. M. Perillo and W. Heinzelman. *Sensor Management, Wireless Sensor Networks*, Kluwer Academic, 2004
9. L. Song and D. Hatzinakos. A cross-layer architecture of wireless sensor networks for target tracking, *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 145–158, 2007
10. Q. Zao and L. Tong. QoS Specific Medium Access Control for Wireless Sensor Networks Fading, *Eighth International Workshop on Signal Processing for Space Communications*, Catania, Italy, July 2003
11. Y. Liu, I. Elhanany, and H. Qi. An Energy-Efficient QoS-Aware Media Access Control Protocol for Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems*, November 2005
12. M.C. Vuran and I.F. Akyildiz. Spatial correlation-based collaborative medium access control in wireless sensor networks, *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 316–329, 2006
13. M. Caccamo, L.Y. Zhang, L. Sha, and G. Buttazzo. An implicit prioritized access protocol for wireless sensor networks. In *Proceedings of IEEE Real-Time Systems Symp.*, Dec. 2002, pp. 39–48
14. S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. “TinyDB: An Acquisitional Query Processing System for Sensor Networks,” in *ACM Transactions on Database Systems*, 2005
15. W.S. Conner, J. Chhabra, M. Yarvis, and L. Krishnamurthy. Experimental Evaluation of Synchronization and Topology Control for In-Building Sensor Network Applications, in *Proceedings of Wireless Sensor Networks and Applications*, San Diego, CA, September 2003
16. J. Beutel. Metrics for sensor network platforms. In *ACM RealWSN'6*, Uppsala, Sweden, June 2006
17. E. Felemban, C. Lee, and E. Ekici. MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks, *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006
18. D. Ganesan et al. Highly resilient, energy efficient multipath routing in wireless sensor networks, *Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2002
19. X. Huang and Y. Fang. Multiconstrained QoS multipath routing in wireless sensor networks, *Wireless Networks Journal*, vol. 14, no. 4, pp. 465–478, 2007
20. N. Jain, D. Madathil, and D. Agrawal. Energy aware multi-path routing for uniform resource utilization in sensor networks. In *International Workshop on Information Processing in Sensor Networks (IPSN)*, April 2003
21. C. Lu et al., RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks, In *Proceedings of the Eighth Real-Time and Embedded Technology and Applications Symposium*, IEEE CS Press, Los Alamitos, CA, 2002
22. B. Karp and H.T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, In *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking*, August 06–11, pp. 243–254, 2000
23. Y.-B. Ko and N.H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, October 25–30, pp. 66–75, 1998

24. J. Elson, Deborah Estrin, Time Synchronization for Wireless Sensor Networks. In Proceedings of the 15th International Parallel & Distributed Processing Symposium, April 23–27, 2001, p. 186
25. I. Aad and C. Castelluccia. Differentiation Mechanisms for IEEE 802.11. IEEE INFOCOM 2001, Anchorage, Alaska, April 20
26. T. He, J.A. Stankovic, C. Lu, and T.F. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks, In Proceedings of International Conference on Distributed Computing Systems (ICDCS '03), Providence, RI, May 2003
27. K. Liu, N. Abu-Ghazaleh, and K.D. Kang. JiTS: Just-in-Time Scheduling for Real-Time Sensor Data Dissemination. In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06), Washington, DC, IEEE Computer Society, Silver Spring, MD, 2006, pp. 42–46
28. Q. Huang, C. Lu, and G.-C. Roman. Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints. International Workshop on Information Processing in Sensor Networks, Palo Alto, CA, April 2003
29. S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong. TAG: A Tiny AGgregation service for Ad-Hoc sensor networks. In Proceedings of the Fifth symposium on Operating Systems Design and Implementation, December 09–11, 2002
30. R. Kumar, S. PalChaudhuri, D. Johnson, and U. Ramachandran. Network Stack Architecture for Future Sensors, Rice University, Computer Science, Technical Report, TR04-447, 2004
31. S. Goel and T. Imielinski. Prediction-based monitoring in sensor networks: Taking lessons from MPEG, ACM SIGCOMM Computer Communication Review, vol. 31, no. 5, October 2001
32. J. Watkinson, MPEG-2, Butterworth-Heinemann, Newton, MA, 1998
33. M.A. Sharaf, J. Beaver, A. Labrinidis, and P.K. Chrysanthis. TiNA: A scheme for temporal coherency-aware in-network aggregation. In Proceedings of the Third ACM International Workshop on Data Engineering for Wireless and Mobile Access, San Diego, CA, September 19–19, 2003
34. S. Santini and K. Romer. An Adaptive Strategy for Quality-Based Data Reduction in Wireless Sensor Networks. In Proceedings of the Third International Conference on Networked Sensing Systems (INSS'06), Chicago, IL, 2006
35. S. Haykin. Least-Mean-Square Adaptive Filters. Edited by S. Haykin, New York: Wiley-Interscience, 2003
36. Embedded WiSeNts. <http://www.embedded-wisents.org/>, December 2006
37. K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for self-organization of a wireless sensor network, IEEE Personal Communications Magazine, vol. 7, no. 5, pp. 16–27, Oct. 2000
38. E. Cheong and J. Liu. galsC: A Language for Event-Driven Embedded Systems. In Proceedings of the Conference on Design, Automation and Test in Europe, March 07–11, 2005, pp. 1050–1055
39. D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler. The nesC Language: A Holistic Approach to Networked Embedded Systems. In Proceedings of Programming Language Design and Implementation (PLDI) 2003, San Diego, CA, June 2003
40. D. Janakiram and R. Venkateswarlu. A Distributed Compositional Language for Wireless Sensor Networks. In Proceedings of IEEE Conference on Enabling Technologies for Smart Appliances (ETSA), Hyderabad, India 2005
41. Srisathapornphat, C. Jaikaeo, and C. Chien-Chung Shen Sensor Information Networking Architecture. International Workshops on Parallel Processing, pp. 23–30, 2000
42. B. Greenstein, E. Kohler, and D. Estrin. A Sensor Network Application Construction Kit (SNACK), In Proceedings of the Second International Conference on Embedded Networked Sensor Systems, November 03–05, 2004
43. I.F. Akyildiz, T. Melodia, and K.R. Chowdhury. A survey on wireless multimedia sensor networks, Computer Networks, vol. 51, 921–960, 2007
44. Y. Gu, Y. Tian, and E. Ekinici. Real-time multimedia processing in video sensor networks, Image Communication, Elsevier Science, vol. 22, no. 3, 2007
45. Y. Wang, R.R. Reibman, and S. Lin. Multiple description coding for video delivery, In Proceedings of the IEEE, vol. 93, no. 1, pp 57–70, January 2005



46. M. Chu, J.E. Reich, and F. Zhao. Distributed attention for large video sensor networks. In Proceedings of the Institute of Defence and Strategic Studies (IDSS), London, UK, February 2004
47. S. Kompella, S. Mao, Y.T. Hou, and H.D. Sherali. Cross-layer optimized multipath routing for video communications in wireless networks, *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 831–840, May 2007
48. B. Krishnamachari, D. Estrin, S.B. Wicker. The Impact of Data Aggregation in Wireless Sensor Networks. In Proceedings of the 22nd International Conference on Distributed Computing Systems, July 02–05, 2002, pp. 575–578