12

Mechanism and Instance: a Research on QoS based on Negotiation and Intervention of Wireless Sensor Networks

Nan Hua[1] and Yi Guo[2]
[1]*Institute of Telecommunication Engineering, Air Force Engineering University China*
[2]*East China University of Science and Technology China*

1. Introduction

The definition of QoS (Quality of Service) varies with the concerned network techniques (wired networks, wireless access networks, wireless Ad hoc networks or wireless sensor networks, etc) and the viewpoint of observation (application level or network level) (Chen & Varshney, 2004; Crawley et al.,1998). The concerned topics of QoS in traditional networks are all end-to-end, and the bandwidth utilization is a core issue of QoS mechanism due to the requirements of multimedia applications. Although there are differences among the specific realization techniques, the research models of QoS are similar and the metrics for evaluating and describing QoS are roughly the same (Chen & Varshney, 2004).

Today, the research on the QoS of traditional networks is mature considerably in theory and practice. In wireless sensor networks (WSN), due to the features such as the limited resource (including energy, bandwidth, cache ability, storage capacity, processing capacity, transmission power, etc), high data redundancy, dynamic topology of network and specific application, the QoS problems are different from that of the traditional networks in the design and implementation. For example, in IP networks, a primary intention of QoS is to ensure that the traffic streams which have different grades or types can get corresponding and predictable transmission services. The grade of service can be classified into best-effort service, differentiated service and guaranteed service. In WSN, because of the unpredictable behavior of edge-to-edge, it is not realistic to provide predictable and reliable transmission service for traffic stream. Hence the QoS of WSN is based on unreliable and best-effort data transmission, but it does not exclude the expression method of traffic (task) stream based priority level. Moreover, WSN reduces the requirements for the packet loss rate to a certain degree; the main concerned issues are no longer the efficient utilization of bandwidth, and the QoS is not always end-to-end.

The researches on QoS mainly involve two aspects: mechanisms and metrics. The classical QoS research results of WSN were summarized by Chen and Shearifi. (Chen & Varshney,

2004; Sharifi et al., 2006). In addition, the issues about QoS of WSN are involved or taken into account in many papers in recent years, while conducting the research on the routing and clustering (topology control) protocol, MAC protocol, as well as application issues, etc (Fapojuwo & Cano-Tinoco, 2009; Hoon & Sung-Gi, 2009; Zytoune et al., 2009; Peng et al., 2008; Chen and Nasser, 2008; Yao et al., 2008; Gelenbe & Ngai, 2008; Navrati et al., 2008; Youn et al., 2007; Zhang et al., 2007; Zhang & Xiong, 2007). The QoS issues involved mainly focus on the instantaneity, fault tolerance capacity and energy consumption of networks, and are studied with the respective research fields of these papers conjointly. All these researches on QoS mentioned above belong to the research field of metrics, these researches neither focus on the QoS mechanism nor discuss the QoS issues of WSN specially and systematically from the basis and architecture. To the best of our knowledge, in the research field of QoS mechanisms of WSN, few distinctive researches are conducted at the present time. In these researches, some QoS schemes based on cross-layer QoS optimization (Cai and Yang, 2007), adaptable mobile agents (Spadoni et al., 2009), cloud model (Liang et al., 2009) and limited service polling discipline analytical model (Aalsalem et al., 2008), and so on, were presented, but are not very mature yet.

In this chapter, we focus our research domain on the mechanisms, the concrete QoS metrics is beyond our discussion scope. In this chapter, we bring forward an Active QoS Mechanism (AQM), the core of it is the negotiation between applications and network and the active intervention for them. On this basis, we conduct a further research, present and realize a common QoS infrastructure as an instance of AQM, named QISM (QoS Infrastructure base on Service and Middleware). The application, state and role oriented QoS optimization scheme, the middleware and service based architecture, the Topic and functional domain based expression method are important characteristics of QISM. Proved by simulation of a typical scenario, QISM has good QoS control ability and flexibility, can support complex applications, and is independent of network architectures.

The rest of chapter is organized as follows. In section 2, we present two QoS levels of WSN and analyze the relationship between the essential problems and QoS. In section 3, we bring forward the concept of AQM, and the working processes, the fundamental of state evaluation and strategy generation are discussed. In section 4, the design philosophy and important characteristics of QISM are studied. In section 5, the infrastructure and realization of QISM are presented and analyzed from four aspects in detail. Then, the simulation results are illustrated in section 6. Finally, we conclude this chapter in section 7.

2. Essential Problems and QoS of WSN

2.1 Three Essential Problems of WSN

We present three essential research problems which should be considered seriously in the applications of WSN through a representative application scenario:

In order to deploy WSN nodes in hostile battlefield or terrible conditions, we normally use airdrop to execute this task. After the nodes bestrewn, it is possible that quite part of them cannot work properly, which leads to heterogeneous distribution of the nodes. Furthermore, it is impossible to supply power when the node energy is exhausted. So, when the network is established, we should face three essential problems as follows:

1) Network Organization
When old nodes invalidated or new nodes joined, the network will be reorganized. Reorganization of network involves many complex processes, such as route rebuilding (the route optimization), topology reconstruction (the selection between the plane architecture and the hierarchical architecture of network, and the transformation from one to another) and task transference (new joined nodes or other working nodes resume the tasks of the disabled nodes), etc.

2) Lifetime of Network and Nodes
To prolong the lifetime of whole network, nodes should work in an energy-efficient way, which includes node dormancy and exchanges of node roles (for example, cluster head, cluster member and router node are three different roles of the nodes, which node acts as which role can be decided through elections and the role of node should alternate periodically). Through these methods, it is mostly possible to average energy consumption of the nodes and ensure the lifetime of key nodes.

3) Quality of Service
We must get tradeoff between lifetime and QoS demand of the network. For example, for the nodes in a lower-density region or executing key tasks, we should find a way to get the necessary tradeoff between application quality and node energy consumption, ensure the achievement of application and the maximum lifetime of network.

2.2 Two QoS Levels of WSN

WSN is a fully distributed network, the QoS of it can be divided into two correlative levels as follows:

1) Network (Application) QoS Level
This level focuses on the whole network, and considers quality of service with a global view of network. The concerned issues involve network organization, network lifetime, and so on. Since Application is a concept correlative with Network, the issue about the analyses of application quality and network state should also be considered in this level.

2) Node (Task) QoS Level
This level focuses on the network nodes, regulates nodes based on the analyses of metrics and data of concrete nodes under the direction of network (application) QoS level, and feeds back data to it for the problem solving of network (application) QoS level. Since Task is a concept correlative with Node, the issue about the analyses of task quality and node state should also be considered in this level.

These two levels of QoS are correlative. For example, the node energy consumption (an issue in node (task) QoS level) is closely related to the network lifetime (an issue in network (application) QoS level), while the energy saving strategy of network (an issue in network (application) QoS level) would affect the lifetime of single node (an issue in node (task) QoS level). The problems in network (application) QoS level have no way to be solved just through the data of some isolated nodes, but the acquisition and analyses of global network situation. The problems in node (task) QoS level generally are the basis of the problems solving of network (application) QoS level, but it is also independent to a certain extent.

2.3 Relationship between Essential Problems and QoS of WSN

Each essential problem of WSN described in 2.1 is not isolated, but is correlative and interact as both cause and effect. Each problem can be divided vertically into two levels: network and node, which is also correlative and affect each other. Hence, we can consider and design a mechanism that could synthetically consider the problems of network organization, lifetime and quality of service of WSN. Above all, this mechanism should associate the regulation in network level with the adjustment in node level and make them become an organic whole, which will guarantee the achievement of applications and prolong the lifetime of network furthest, meanwhile the requirement of application for network behavior is satisfied as far as possible. As discussed in 2.2, the QoS of WSN is composed of two correlative levels: network and node, so we have reason to believe that a specially designed QoS mechanism is a good way to solve the problems mentioned above.

3. Active QoS Mechanism

Generally speaking, the core of QoS mechanism in traditional networks (for example IP networks) is that how to satisfy the requirements of applications for network capability through given methods and mechanisms. The basic process of it can be described that network try its best to satisfy the requirement proposed by application; if the requirement cannot be satisfied, the network will degrade the quality of service and feeds back it to the user. We call this traditional QoS mechanism.

However, the traditional QoS mechanism will bring some problems in WSN. For example, under the circumstance of battlefield supervision application, traditional QoS mechanism will terminate the application and return errors when the object node executing key tasks or the cluster head is disabled. But actually, the application can be achieved if we reorganize network in right time and transfer the tasks in disable nodes to other normal nodes properly.

3.1 Theory of AQM

The key to solving problems mentioned above is that a feedback and negotiation mechanism must be established between the applications and network when the support of network to applications or / and the applications demand to network is / are changed. This mechanism regulates the network and applications under certain strategies dynamically, makes the applications adapt to network and network support applications furthest, and improves the support ability of WSN to applications and adaptability of applications to WSN. This feedback and negotiation mechanism between network and applications is named Active QoS Mechanism (AQM) by us.

The key of AQM is the process of active intervention for applications and network. This process is built on the analysis and evaluation for the states of applications and network, which involves two aspects: the regulation of applications to network and the reaction of network to applications. Collecting information from applications and network, and analyzing / evaluating the states of them with the information collected is the foundation of AQM.

This mechanism is not necessary in traditional networks, but it is directly related to the lifetime of applications and network in WSN. The fundamental reason of this lies in the unreliable network elements, the instability and resource-constrained nature of WSN.

3.2 Working Process

The working process of AQM involves four phases: initialization phase, surveillance phase, negotiation phase and regulation phase. The relationship of these phases is illustrated in Fig. 1. Besides, the relationship of application, network, AQM and main output in each phase are presented in Fig. 2.

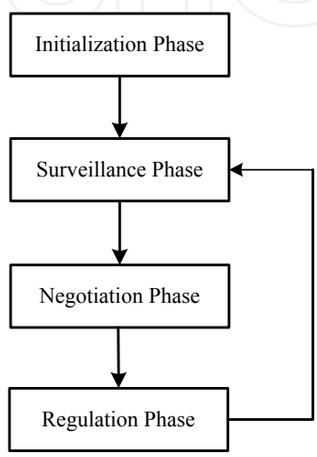


Fig. 1. Four phases in working processes of AQM

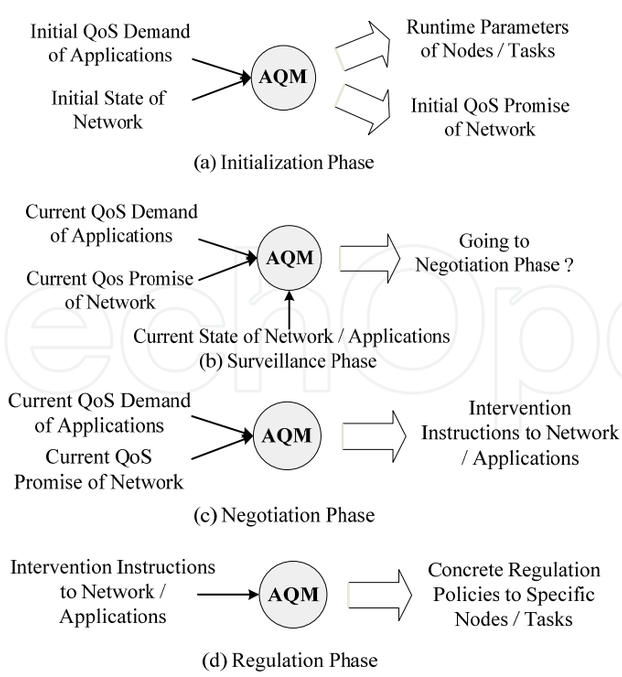


Fig. 2. Main input and output of AQM in different working processes

1) Initialization Phase
Combined with the initialization process of network, AQM generates the initial QoS promise according to the requirements of applications for QoS and the initial state of network, and sets the runtime parameters of nodes and tasks according to the initial QoS promise.

2) Surveillance Phase
AQM traces the state of applications and network constantly, and monitors the QoS demand of applications. When there is a conflict between current QoS demand of applications and current QoS promise of network, AQM goes to negotiation phase.

3) Negotiation Phase
Through AQM, a negotiation and tradeoff is achieved according to the QoS demand of applications and the QoS promise of network, and then the intervention instructions to the network and / or applications are generated. AQM goes to regulation phase.

4) Regulation Phase
According to the intervention instructions to the network and / or applications, the concrete regulation policies to specific nodes and / or tasks are generated and the runtime parameters of specific nodes and / or tasks are modified by AQM, AQM goes to surveillance phase.

3.3 State Evaluation and Strategy Generation

AQM produces the evaluation to the state of applications and network, generates regulation strategy to applications (network) and tasks (nodes). This is a process of analyzing and optimizing applications and network according to the states of them combining with the requirement of applications, this process is application, state and role oriented. We can regard state evaluation and strategy generation function of AQM as a black box, which owns a predefined method set. The input of this black box is correlative with the application demand to network, current application state, current and previous network state and current QoS promise of network. The output of it involves the intervention instructions to network and / or applications, the concrete regulation policies to specific nodes and / or tasks (in the form of runtime parameters), as shown in Fig. 3.

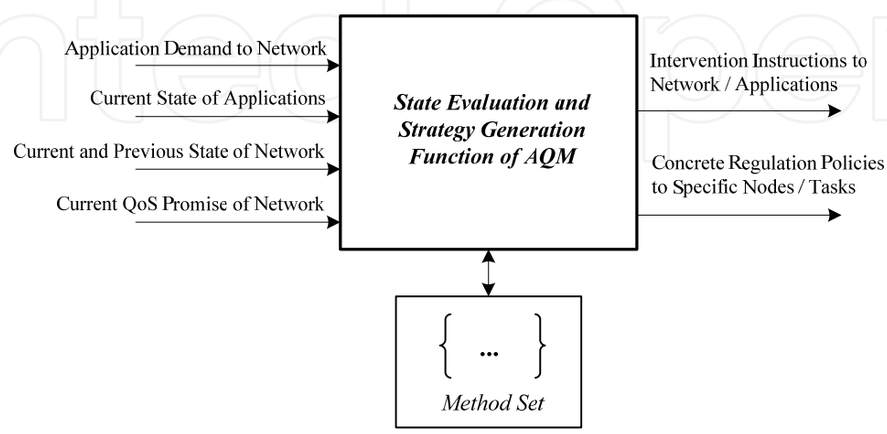


Fig. 3. Fundamental of state evaluation and strategy generation of AQM

4. QISM: an Instance of AQM

From this section, we design and realize a common QoS infrastructure as an instance of AQM, named QISM (QoS Infrastructure base on Service and Middleware) by us. The design philosophy of QISM is as follows:

4.1 Application, State and Role oriented QoS Optimization Scheme

The core of AQM is negotiation and intervention, which is based on the analyses of previous accomplishment quality of applications, current requirements of applications for the quality of service, the current and previous states of network, as well as the current service promise of network. These analyses are based on applications, states and roles. Since the application, state and role are time variant in WSN, these analyses are dynamic too.

1) Application-oriented

The main idea is to distinguish task streams, and different kind of task stream should acquire the support of different QoS in different time. This assignment of QoS should consider the previous and current states of network. Not only the distribution according to need but also the possible carrying capacity of network should be considered.

2) State-oriented

The previous and current states of network (applications) and nodes (tasks) should be considered when negotiation and intervention is proceeding; even previous data packets should be analyzed if necessary.

3) Role-oriented

The Regulations to network and nodes should consider the status and functions of nodes in current network. For example, the nodes that carry out a key sensing task should avoid becoming cluster head or router node in order to save energy and prolong its lifetime.

4.2 Middleware and Service based Architecture

Currently, there are close coupling between software and hardware, as well as applications and operating system of WSN, which has brought inconvenience for the task transference as well as the development and adjustment of hardware and software. Middleware is a software layer, which can provide services for various applications and enable different application processes to communicate via network under the circumstances of shielding difference among platforms. Through the middleware, it is convenient to provide standard system services, support and coordinate multiple runtime environments, and efficiently utilize the resource of network. The architecture of QISM based on middleware is shown in Fig.4

When an application is being performed, the application is decomposed into relatively independent tasks firstly, and then the services are abstracted from tasks. The system requests and subscribes the services, gets the required data and completes the requested functionality. Service is a concept about “set”, it is a logical abstraction of homogeneous tasks from the viewpoint of network. Service indicates “what to do” and implies the functional domains related with service. Task is concept about “individual”, including not only “what to do” but also “how to do”. For instance, for the service such as “temperature”, many nodes possibly support the task of temperature acquisition. But how to acquire, i.e.

"how to do", such as the thresholds and sampling frequency setting, is related with the tasks and nodes. Different nodes probably have different parameter values, which are decided by their runtime parameters. The relationship between services and tasks is shown in Fig. 5.

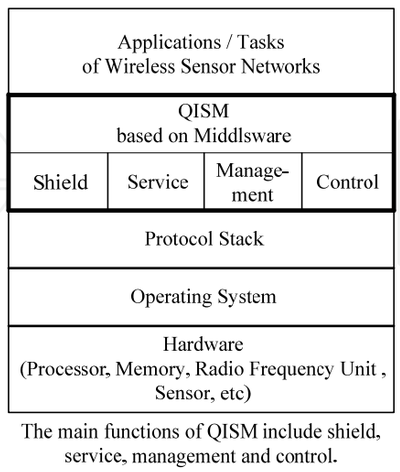


Fig. 4. Architecture of QISM based on middleware

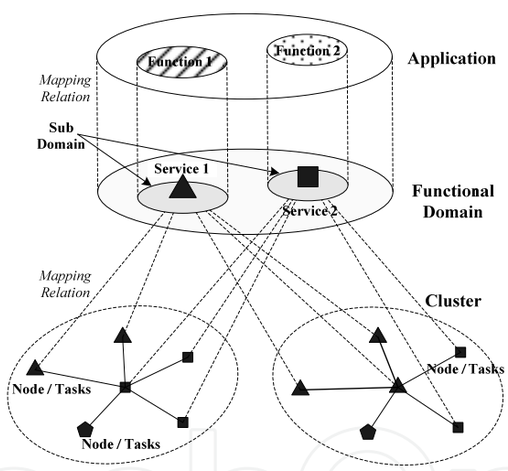


Fig. 5. Relationships among application, services, tasks and functional domain

4.3 *Topic* and Functional Domain based Expression Method

Topic is always associated with the concept application, an application can have more than one *Topic*, and a *Topic* can be associated with multiple applications. The syntax of *Topic* is defined as follows:

Topic < AppName > [< AppName > [...]] < TpStyle > < TpDesp > [< TpDesp > [...]]

where *AppName* is the name of an application and unique in the network, which is the distinction from other *Topic*s of applications. The style of *Topic* is identified by *TpStyle* and *TpDesp* is the specific description of the content of the *Topic*. *TpDesp* can be Interests and Events of WSN, or other control information related with the application, such as various

commands or messages. The control information is denoted as *SysCtrlInfo*. Different from Interest and Event, *Topic* is based on the application (network) level while Interest and Event is in the task (node) level.

Functional domain is a node set that involves all nodes which provide all kinds of services requested by a specific application, no matter whether the tasks of the nodes are working or not. The node subset that provides different services is a sub domain of the functional domain of the specific application. Functional domain is related with specific application and associated with specific Interest and Event. For example, for the application of fire monitoring, if we wants to acquire the data of temperature and smoke fume, the functional domain related with fire alarm application is the node set that involves temperature and smoke sensor nodes, the sub domain of it are the node subset that involves temperature sensor nodes and the node subset that involves smoke sensor nodes only, respectively associated with the Interest and Event of temperature and with that of smoke.

Functional domain is presented from the viewpoint of application and is unrelated with the architecture models that the network uses. In the hierarchical architecture model of network, such as cluster, a functional domain or its sub domains can cover several clusters. The relationships among application, services, tasks, and functional domain are shown in Fig. 5.

5. Infrastructure and Realization of QISM

5.1 Architecture and Function

According to the discussion in 4.2, QISM is base on middleware and is a software layer that located between the protocol stack and applications, communicating with application / task and protocol stack through standard API. QISM is composed of six modules: application analysis, application / task regulation and control, strategy generation / analysis, state analysis, service management, *Topic* generation / resolving. The hierarchical relationship of the above-mentioned modules is shown in Fig. 6. Each module lies in sink and (or) sensor node, as shown in Table 1.

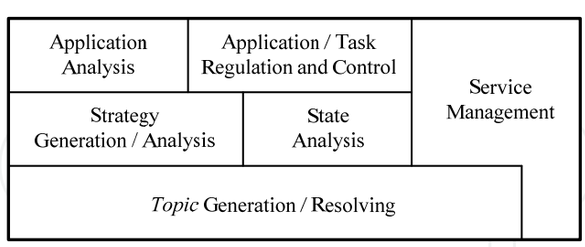


Fig. 6. Hierarchical architecture of QISM

Module Name		Location	Function
Application Analysis		Sink	Decomposing application into tasks according to the description of application, and determining whether the tasks are supported by existing available services through Service Management Module. If necessary, indexing and subscribing related services through Service Management Module.
Application / Task Regulation and Control	Application Regulation and Control	Sink	Analyzing implementation status depending on functional domain states and services states, evaluating whether or not the network supports application, and completing application regulation and control.
	Task Regulation and Control	Sensor Node	Completing task regulation and control through setting runtime parameters of task.
Strategy Generation / Analysis	Strategy Generation	Sink	Generating runtime parameters of tasks according to application requirements as well as current application and node state in the states library.
	Strategy Analysis	Sensor Node	Resolving runtime parameters, determining whether current node is in specific functional domain.
State Analysis		Sink	Analyzing task implementation status, determining functional domain and service state, evaluating network state, maintaining the states library.
Service Management		Sink, Sensor Node	Realizing service publication and subscription mechanism, and functions of service discovery, indexing and maintenance.
Topic Generation / Resolving		Sink, Sensor Node	Packing and unpacking *Topic*.

Table 1. Main modules and functions of QISM

5.2 Service Management

The functions of service management of QISM, which consist of publication, subscription, inquiry, index and maintenance of services, are implemented through Service Management Module. The service publication and subscription mechanism is the basis of QISM and the main usage mode of service, where the task side (sensor node) publishing services initiatively and the application side (sink) subscribing and using them. Furthermore, the service inquiry and index mechanism provides the methods that can acquire the state of service, and the methods of requesting and activating service from the application side. The function of service maintenance is used in recording and maintaining the services which are published in the network already, and the function is realized in sink and sensor nodes locally. In sink, table *TASvc* and *TOSvc* have the records of current available services and subscribed services respectively; in sensor node, the subscribers of node services are recorded in table *TSvcOd*. Subscription, inquiry and index function are implemented in the sink, publication function is done in sensor nodes, maintenance function both in the sink and sensor nodes.

The processes of service publishing, subscribing, inquiring and indexing in QISM are illustrated as Fig. 7.

1) Publication and subscription of service

Publication and subscription of service involve two kinds of messages: *MsgSvc* and *MsgSvcOd,* their syntaxes are defined as follows:

MsgSvc < SvcName > < SvcPrvdID > [< SvcDesp >]

MsgSvcOd <AppName> < SinkID > < SvcName > [< SvcPrvdID > < SvcDesp >]

where *SvcName* is the name of service; *SvcPrvdID* and *SinkID* are the IDs of the service provider and the sink respectively, which can be addresses, domains or coordinates and so on; *SvcDesp* is the description of the service.

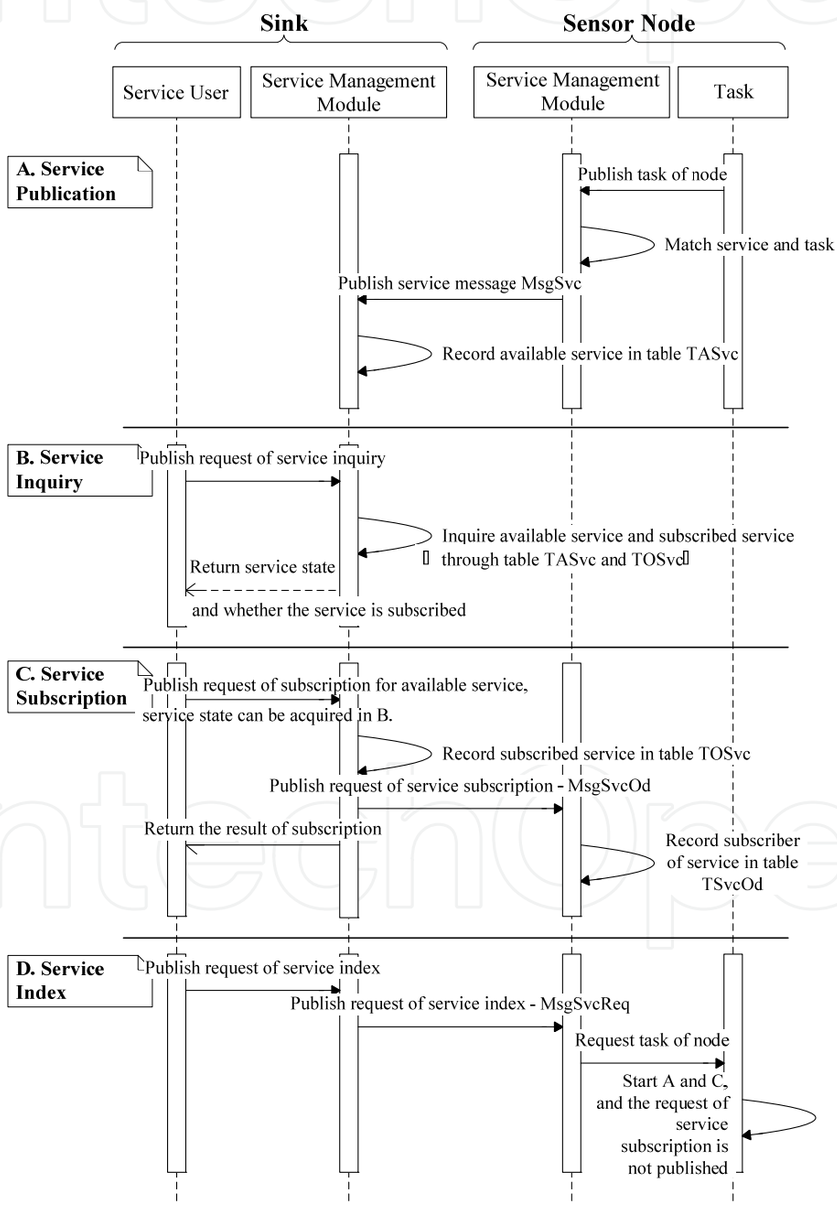


Fig. 7. Processes of service publishing, subscribing, inquiring and indexing in QISM

After the network deployed, sensor node will publish and broadcast the tasks (which can be performed by it) through *MsgSvc* in the form of service; after received by sink, the services are saved in *TASvc* and determined whether to be subscribed according to the requirements of application. If the service is useful, the sink sends message *MsgSvcOd* to *SvcPrvdID* to subscribe it, and records the subscribed service in *TOSvc*. After the sensor node receives *MsgSvOd* which is sent to it, it records the subscriber in *TSvcOd*. Based on the consideration of resource saving and network survivability, sensor node dose not record *MsgSvc*s that are sent by other nodes.

If *SvcPrvdId* is specified in *MsgSvcOd*, which means the sink subscribes the service that is provided by specific sensor node; otherwise, which means the sink subscribes all the same services that are provided by all nodes in the network. When sending service data, sensor node will specify the data receiver. In the case of multiple sinks, the sink that did not subscribe the service, will discard service data directly after the service data is received.

2) Inquiry and index of service

The state of service is either *Available* or *Unavailable*; the state of specific service can be acquired through inquiring *TASvc* in sink. If a service is available, it can be used through subscribing. Otherwise, it means that the service has not been published by any nodes yet. In this case, if we want to use the service, we should start the service index mechanism in sink. The sink sends message *MsgSvcReq* to the network firstly, then the sensor nodes that are capable of providing the service publish the service, finally the sink subscribes the service and uses it. The syntax of *MsgSvcReq* is defined as follows, where *SvcReg* stands for the region where the service is located.

MsgSvcReq < SvcName > < SinkID > [< SvcReg > < SvcDesp >]

3) Maintenance of service

The service maintenance functions of QISM mainly include the table maintenance and update of *TASvc*, *TOSvc* and *TSvcOd*, as well as service cancelling and unsubscribing. When sensor node is unable to provide services, such as under the circumstances that sensor is damaged, *MsgSvcFail* is broadcasted and *TSvcOd* is cleared by the sensor node. After the sink receives *MsgSvcFail*, *TASvc* and *TOSvc* (if the service is subscribed already) are updated in order to cancel the service. The syntax of *MsgSvcFail* is defined as follows:

MsgSvcFail < SvcName > < SvcPrvdID > [< SvcDesp >]

When the application no longer needs a specific service, the sink sends message *MsgSvcCancel*, and deletes the corresponding service from *TOSvc*. The sensor node that provides the service maintains a user counter, and when it receives *MsgSvcCancel*, the corresponding counter of the service is decreased by one and *TSvcOd* is updated at the same time. When the counter is reduced to 0, the sensor node broadcasts *MsgSvcFail*. The syntax of *MsgSvcCancel* is defined as follows:

MsgSvcCancel < SvcName > < SinkID > [< SvcPrvdID > < SvcDesp >]

It should be noted that the service publication only means that sensor node has the ability of carrying out a task, but when to start or to terminate the task, as well as how to implement the task depends on the runtime parameters. More specifically, under the control of the application, task-performing is achieved through the built-in mechanism of QISM by

correlative modules generating, sending and implementing the runtime parameters, and it is unrelated with service management module. Moreover, the runtime parameters of tasks are not saved in service management module. Besides, the above-mentioned messages related with service, are sent directly through network protocol stack by service management module.

5.3 Basic Working Process

From the viewpoint of the operator of QISM, QISM includes two basic working processes: dynamic adjustment of application and active regulation of task, as shown in Fig. 8. Both are associated closely and reciprocal causation, as a unified organic whole.

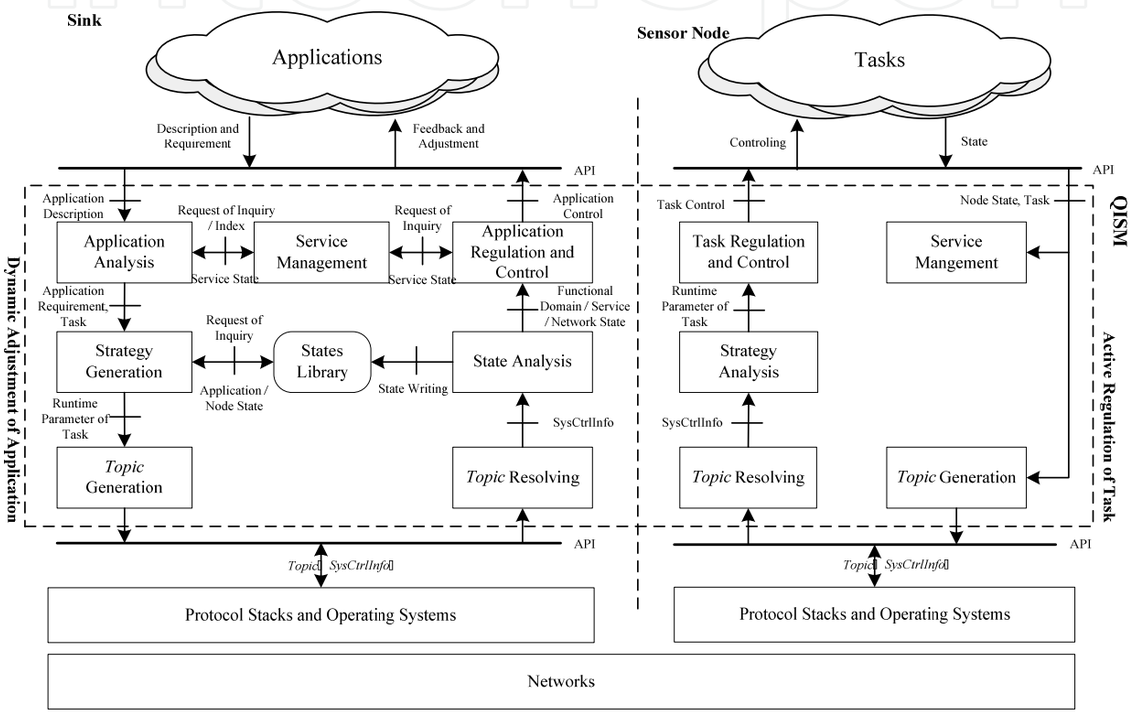


Fig. 8. Data stream of QISM

QISM first completes the service subscription process according to the description and requirement of the application, and then generates the runtime parameters. Afterwards, QISM publishes the runtime parameters of the tasks, and starts the processes of regulations of application (network) and task (node). In sensor node side, QISM intervenes the execution of tasks by setting runtime parameters of tasks, and feeds back the states of nodes and tasks to sink; QISM regulates the application after state analysis process, and then generates the new requirements and (or) descriptions of the application. Such a repetition will form a closed loop until the ends of tasks.

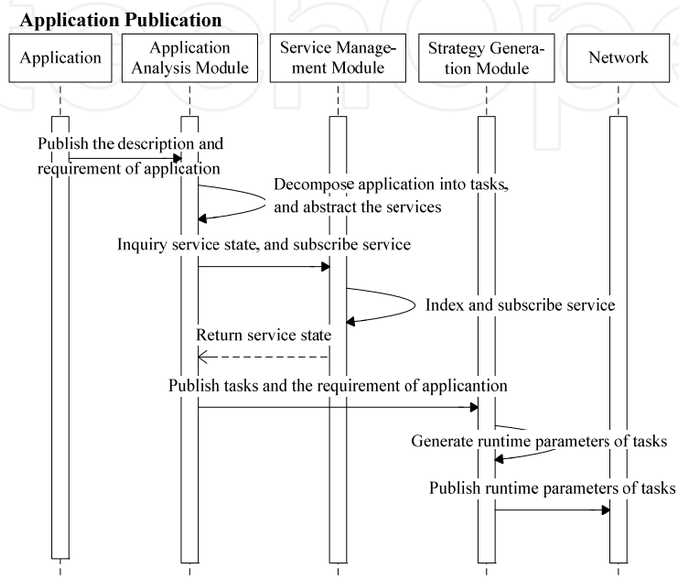
It should be noted that, we do not reflect the processing methods and flow direction of the Interest and Event in Fig. 8 and in the following discussion. In fact, since Interest and Event is a kind of organization and representation method of data, the requirements and descriptions of application may contain the content of Interest, and the states that fed back to QISM from tasks may include a part of data of Event. Transmission of Interest and Event

can be implemented by *Topic* mechanism or other methods. A detailed discussion of Interest and Event is beyond the scope of this chapter.

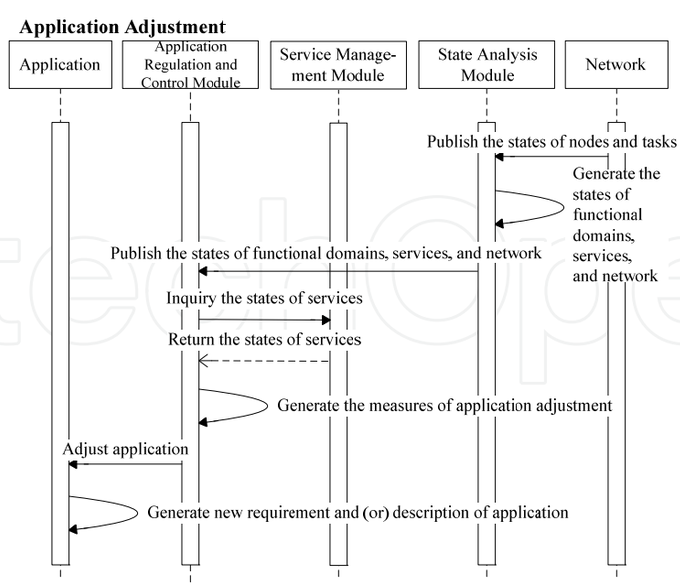
1) Dynamic adjustment of application

Dynamic adjustment of application, whose operator is sink, consists of two processes: application publication and application adjustment, as shown in Fig. 9(a) and (b).

- *Application publication (the downlink process from application to network)*



(a) Application publication



(b) Application adjustment

Fig. 9. Basic working process of QISM - Dynamic adjustment of application

Firstly, QISM decomposes application into several independent tasks according to the description of the application, and abstracts the service corresponding to the tasks. For example, for the fire monitoring application, temperature monitoring and smoke monitoring are two tasks that need to be accomplished; in the node level, the services that are provided by the nodes with the ability of sensing temperature and sensing smoke are temperature sensor service and smoke sensor service respectively. The division, abstraction and correspondence of task and service, is based on the pre-defined rules, which are fixed when the network is deployed.

Secondly, QISM subscribes services. If the services are available, they can be used after subscription; if not available, they can be activated by service index mechanism and then be subscribed. Eventually, all the services required by the application should be available; otherwise, QISM will terminate the application and cancel all the tasks.

Thirdly, QISM generates the runtime parameters of the tasks according to the request of application. The runtime parameters, including functional domain, sampling frequency, thresholds and so on, have great influence on the service quality and execution manner of tasks. In addition, energy strategy is also an essential parameter. The death of some important nodes whose functions are irreplaceable, such as the cluster headers in hierarchical structure, the key routing nodes in multi-hop routing, the key sensor nodes, and so on, may cause the failure of the application or the collapse of the network. So the energy strategy should be established in order to prolong the lifetime of nodes.

Finally, QISM publishes the runtime parameters of tasks to the network in terms of *Topic* (*SysCtrlInfo*), for sensor node receiving and performing.

- *Application adjustment (the uplink process from network to application)*

The *Topic* (*SysCtrlInfo*) received by sink from network includes the current state information of tasks and nodes; its specific content is determined by the pre-defined rules and is different with different tasks. The above-mentioned state information is the basis of application adjustment.

Firstly, QISM confirms that *SysCtrlInfo* is for this application (sink) through resolving the domain of *Topic AppName*, for there are multiple applications (multiple sinks) in the network probably.

Secondly, the state information of a single node is transformed into measurable QoS metrics, and on this basis, the state of functional domains and that of services are generated and the network state is evaluated. The related QoS metrics consist of network delay, packet loss rate, data reliability of node, node lifetime, node energy consumption per bit, packet transmission delay of node, invalid packet rate of node and node remnant energy, etc.

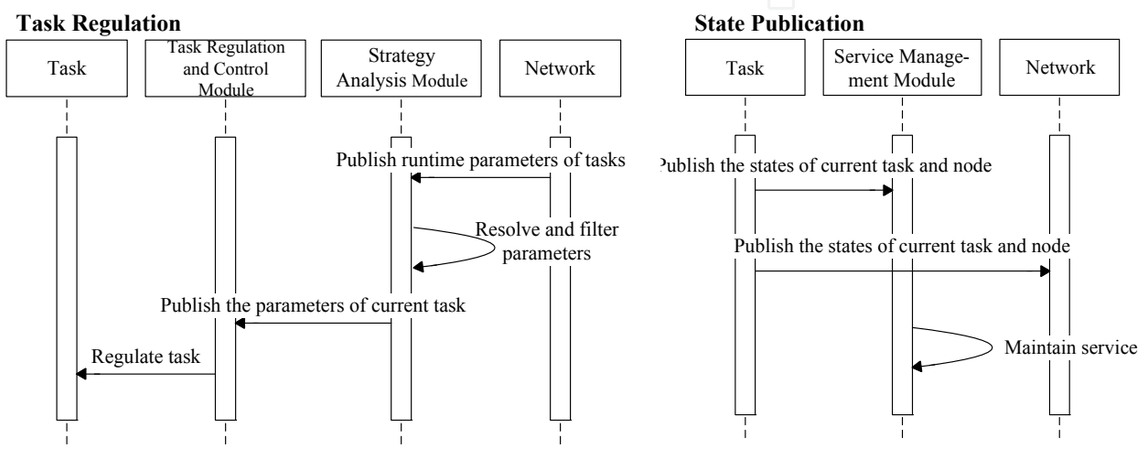
Finally, QISM generates adjustment measures (i.e. intervention instructions to network / applications) for application and informs application to perform, based on the state analysis results, current states of functional domain / service / network and current requirements of application. Application adjustment is faced to functional domain, network and service, not single node and its tasks, though its basis is the information collection and analysis of single node and its tasks. The measures of application adjustment include resuming application, pausing application, resuming application after adjustment, ceasing application, etc.

2) Active regulation of task

Active regulation of task, whose operator are sensor nodes, consists of two processes: task regulation and state publication, as shown in Fig. 10(a) and (b).

- *Task regulation (the uplink process from network to task)*

In sensor node side, *Topic* (*SysCtrlInfo*) received from network consists of the requirements of application for task in the form of runtime parameters of task (i.e. regulation policies to specific nodes / tasks) sent from sink. First of all, QISM confirms that *SysCtrlInfo* is for the functional domain where current node is located through resolving the domain of *Topic AppName*. And then, QISM completes task regulation by setting runtime parameters of the task.



(a) Task regulation (b) State publication

Fig. 10. Basic working process of QISM - Active regulation of task

- *State publication (the downlink process from task to network)*

During the implementation of task, sensor node needs to inform QISM of the current task state (such as whether the task is completed or not, the implementation progress of task) and node state (such as working state of sensor, remnant energy of node). On the one hand, QISM adjusts current services of node according to this, e.g. service is canceled when sensor node is disabled; on the other hand, QISM sends the states to related sink through network for the preparation of state evaluation.

5.4 Task (Node) Refactoring

Through the generation of concrete regulation policies to specific nodes and tasks based on the intervention instructions to applications and network, QISM realizes the task and node refactoring by means of resetting the runtime parameters of specific tasks and nodes. The so-called refactoring means that the functions and performance of tasks and nodes are modified through the reset of runtime parameters of them, which leads to the change of the support ability of network to applications and the QoS demand of applications to network.

The more ideal methods for the implementation of task (node) refactoring involve three schemes as follows, but the concrete implementation method in QISM should be studied more deeply in our further research:

1) Self-adaptive Adjustment of Protocol Architecture
The protocol stack involves several components (protocol elements) which are served for different purposes or applications and have different performances and functional characteristics. When external conditions are changed, the QISM selects and applies proper the protocol element automatically.

2) Software Component Technology
Component is a kind of reusable software element which can be used to construct other software. Software component technology is an object-oriented technical system, which builds applications through the combination of different components and involves a series of correlative operations and services. The core of it is the concept of PnP (Plug and Play) soft component that can work immediately after it is embedded.

3) Downloading and Updating of Protocol and Application
QISM downloads new protocols and updating programs dynamically and on demand from the base station (for example the sink). This method is more flexible but need the coordination with the base station or service center.

6. Simulation and Analysis

QISM has a complex active regulation process for application and task, and its specific logics, including application analysis, application / task regulation and control, strategy generation / analysis, state analysis and service management, etc, depend on specific application and specific realization of system. So we only prove the feasibility of QISM through the simulation for fire monitoring application below.

In fire monitoring application, the network consists of temperature sensor nodes and smoke sensor nodes, crossly deployed in the adjacent regions A and B, as shown in Fig. 11. After the network is deployed, system performs the tasks of temperature and smoke sensing on the support of QISM.

We used ns2 v2.27 to simulate the above scenarios with Linux Red Hat 9. Thirty-six static nodes deployed uniformly in a grid-like plane scene, the temperature sensor nodes and smoke sensor nodes were crossly deployed. The clustering algorithm was DSCO (Hua & Shi, 2007) and cluster head did not alternate. The protocol of MAC layer was 802.11b, Interface Queue (IFQ) length was 50, and Two-ray Ground Reflection was as wireless transmission model. To be brief and without loss of generality, the single-hop communication was adopted between the cluster head and sink.

After cluster organization is completed, the simulation uses the following logic to control and regulate the application and network:

Logic 1: Service publication. Node publishes temperature and smoke service to sink through cluster head.

Logic 2: Application publication, service decomposition and service subscription. Application (sink) subscribes the temperature service *Svc_Tmp* and smoke service *Svc_Fg* of nodes in region A through QISM.

Logic 3: Task runtime parameters generation and task control. The nodes in region A are activated by QISM through dispatching the task runtime parameters (such as sampling frequency f_s) to them, as shown in Fig. 11(a).

Logic 4: Node state and service state publication. Nodes in region A report current node states (such as remnant energy E_r) to QISM meanwhile they feed back the sensing data (such as temperature and smoke concentration) to application through sink.

Logic 5: State analysis of task and node, application active regulation, task regulation and control. QISM ceases the data acquisition task in region A according to pre-defined logics when the energy of 50% nodes decrease to $E_r/3$, and subscribes services *Svc_Tmp* and *Svc_Fg* of region B. The nodes in region B are activated and replace the work of nodes in region A, as shown in Fig. 11(b). Then logic 1-4 are repeated, where nodes in region A is replaced by nodes in region B.

An important reason for designing logic 5 is to prove that active regulation of QISM for application and service can effectively prolong the lifetime of network and application. The results of simulation shows, in the above simple working model based on energy, the lifetime of cluster members are longer than that of members which do not use QISM (all deployed nodes working synchronously) by 30%. The longer lifetime of node is, the longer lifetime of network and application is.

It should be noted that in the above-mentioned simulation, we have not considered the lifetime of cluster head. Energy consumption of cluster heads can be averaged to prolong its lifetime through dynamic alternating cluster head in cluster organization algorithm (Hua & Shi, 2007). The study on dynamic cluster organization is beyond the scope of this chapter.