2018 International Conference on Sensor Network and Computer Engineering (ICSNCE 2018)

The Design of QoS Guarantee Strategy Framework for Networked Control System

Yang Weixia
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, Shaanxi, China
e-mail: 1195862787@qq.com

Xu Fei[a], Wang Shaochang[b]
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, Shaanxi, China
e-mail: [a]29112462@qq.com
[b]1163401315@qq..com

***Abstract*—For networked control systems such as packet loss and delay of the basic problems, this article launches the research from the perspective of network scheduling optimization, build the QoS semantic model and service model for networked control, puts forward the multi-level QoS guarantee strategy deployment algorithm and QoS, QoS access algorithm components; On a class of networked control middleware prototype system of network resources to meet of components and component deployment simulation experiment, the results show that by properly adjusting the bandwidth of the task node and priority, and controlling the efficient transmission of information, so as to shorten the network time delay, improve the network performance, increase the efficiency of the controller.**

Keywords-Networked Control; Middleware; Resources to Meet; Components Deployed

I. INTRODUCTION

The networked control system brings many advantages to the control system through network resource sharing, and introduces new challenges to the control circle. The complexity of the network control system is introduced by control system in the network of the characteristics of decision, such as network induced delay, packet loss, single packet of data and multiple packet transmission, packet random sequence and network scheduling problem, this article called "web services digital constraint" [1].Network control applications require the use of heterogeneous, diverse resources, in general use of these resources sharing, more requires collaboration between the various tasks, resource scheduling, task execution is complicated. The current network infrastructure and are based on "best effort" basis, quality of service is not fully guaranteed, multiple services collaboration use resources, performance and efficiency is also easy to cause the drop. Addressing these issues requires the introduction of the QoS mechanism. Ian Foster and other scholars have proposed the QoS framework based on the Globus Toolkit -- the Globus Architecture for Reservation and Allocation, GARA [2][3], which provides the middleware core layer and the end-to-end QoS guarantee for the service layer. GARA use "gramm for resource management, defines the generic description mechanism for all kinds of heterogeneous resources and provide unified control interface, but the lack of extensibility, implementation architecture can't applicable to the general field of networked control request. Professor at the university of Cardiff Rashid QoS management framework - G - QoSM [4], the use of the resources of QoS information registration, discovery and selection of reserve resources, resource allocation and task scheduling adaptive adjustment, the framework is the complete lack of QoS semantic modeling ability, change can not be very good to the business environment. Zeng et al. [5][6] proposed the service composition middleware based on maximization of user satisfaction, and gave two strategies for local task-level service selection and global collaborative allocation. The local service chooses to use the metric and weighted two steps to measure the service in the multidimensional QoS attribute. Literature [7] points out that the heterogeneous network environment put forward requirements definition scheme based on a known as the elastic constraints, the multi-objective problem is modeled as a compound option of knapsack problem, based on dynamic programming algorithm for dynamic constraint, but it is difficult to clearly determine the application of customers' needs.S.S hakkottai et al. [8] in Ad Hoc wireless self-organizing network research, puts forward the idea of "cross layer design" in comprehensive considers the relationship of the protocol stack layers and retain the original layered protocol stack structure, on the basis of breaking the TCP/IP network hierarchy middle layer and the communication between the limit, supplement and designed a new type of network protocol stack, allowing interaction between the layers of communication protocol stacks. Xian university of electronic science and technology of yi-lin chang and others [9] in the cross layer design, on the basis of the five layers of typical network architecture was proposed and "cross-layer perception" concept, will be with the idea of cross-layer design, promotion to the network on the DE facto standard five levels.26/5000 Most current network protocol implementations provide only partial QoS functionality. Networked control systems is to run over in the basic network services, the simple network QoS mechanism can guarantee the QoS requirements, and the existing QoS guarantee strategy are based on "the best" service mechanism, and cannot provide components oriented middleware platform support.

This chapter first analysis of networked control environment context factors, using the method of side oriented QoS based semantic model of real-time system, reconstruction of networked control system service quality in the hierarchical structure and establish the corresponding management model, by extending the object management group interface definition language, will establish the QoS management model integrating into the kernel of the middleware services, and construct the field oriented control service QoS management framework, finally through the simulation experiments verify the QoS management mechanism of effectiveness.

The structure of this paper is as follows: section 2 introduces the QoS semantic model and QoS service model for network-oriented control; Section 3 introduces QoS scheduling framework design; Section 4 introduces the QoS component access algorithm and QoS component deployment algorithm. Section 5 carries out the simulation experiment to verify the above strategy framework and the effectiveness of the algorithm.

II. QOS SEMANTIC MODEL

This section presents a QoS semantic model, its purpose is to establish an open, extensible to define, QoS data exchange and interoperation model, does not allow the presentation logic to QoS number as the QoS parameters, QoS management level, the change of QoS management mechanism and fundamental changes, decouple the presentation logic and processing logic, so as to better adapt to the various business requirements under the networked control systems. The semantic model is composed of QoS model and QoS management model.

A. The metamodel

QoS metamodel defines QoS said system required by the atomic data structure, it said in the QoS system plays a basic structural role, not only to enhance the QoS rigor of said system, and lay a foundation for the expansion of the QoS definition. The QoS model representation system is shown in figure 1 below.

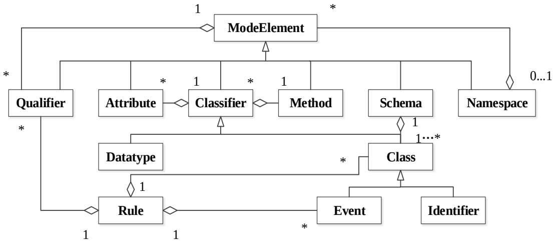

Figure 1. QoS meta model structure

- Model Root Element is the Root Element of the inheritance hierarchy in the metamodel, which is the most basic abstract block.
- Identifier is the string that the model element identifies itself, which is unique in the entire namespace.
- category : abstract building blocks used to derive data types and class 2 objects. It is an aggregation of methods and properties.
- DataType which defines the basic data types used to describe QoS, construct data types, template data types, and custom data types.
- Class: an abstraction of the collection of objects of the same behavior, attributes, relationships, and semantics.
- Schema: a framework consisting of one or more classes.
- Qualifier is the specification and constraint of elements such as class, attribute and its outline.
- Event can be located in time and space and can take place with practical significance.
- Rule is a policy used to indicate a class's status migration when an event occurs.

B. QoS management model

The control application is a set of application artifacts with QoS characteristics. The component instance is the physical unit that is deployed to the instantiation in the component framework, and the runtime component instance can be represented as a process, thread, or other autonomous running unit in the component container environment. An application Configuration Graph, which is called a QoS component, is made up of abstract examples.

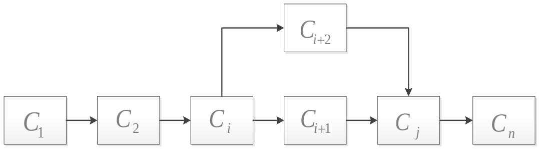

Figure 2. QoS semantic component application configuration diagram

From the component instance to the directed edge, where the output is input. The QoS feature of the component is shown as follows.

$$QoS_{Component} = (QoS_{demand}, Res, QoS_{provide})$$

$$QoS_{demand} = [q_1^{dem}, q_2^{dem}, ..., q_m^{dem}], q_i^{dem} \in double, i = 1, ..., m$$

$$Res = [q_1^{res}, q_2^{res}, ..., q_k^{res}], q_i^{res} \in double, i = 1, ..., k$$

$$QoS_{provide} = [q_1^{pro}, q_2^{pro}, ..., q_n^{pro}], q_i^{pro} \in double, i = 1, ..., n$$

In the above equation, the values of QoS attribute values are all real types, in which the QoS attribute list is represented and required by the component instance respectively. The other is the list of system resources occupied by the component instance runtime, which is provided by the system resource management service.

III. QOS STRATEGY FRAMEWORK

A. Frame structure

This section is presented based on the strategy of QoS monitoring and the safeguard mechanism, it is running the QoS framework core components, not only for the resource service management and the operation of the communication network to provide performance guarantee, also the QoS parameters of dynamic monitoring in operation, to provide QoS strategy execution. Resources service according to the requirements of the performance of communication network, and these requirements is mapped to a communication network QoS standards, use appropriate protocol in communication network and QoS strategies for resource management to provide satisfactory service data transmission performance. The framework is shown in figure 3 below.

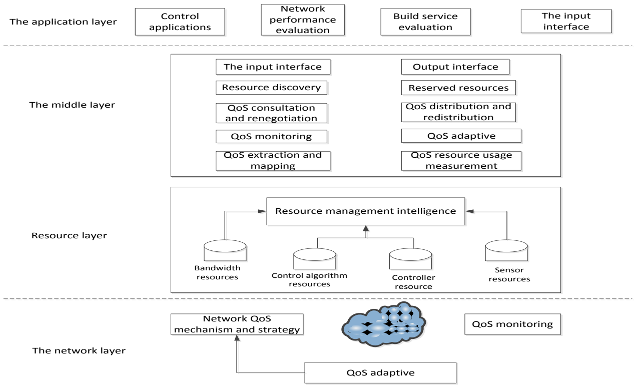

Figure 3. QoS management framework

B. *QoS requirements mapping mechanism*

To be able to provide effective QoS support for control applications, you must be able to define clear QoS requirements from the user's expectations. Here, the concept of Group of Work (GoW) is presented, representing the minimum working granularity unit that can be processed by the resource service. An application can usually consist of several working groups, each of which can be processed by the resource service according to the corresponding QoS parameters. The process of extracting explicit QoS characteristics and requirements from the control application is shown in the figure below.

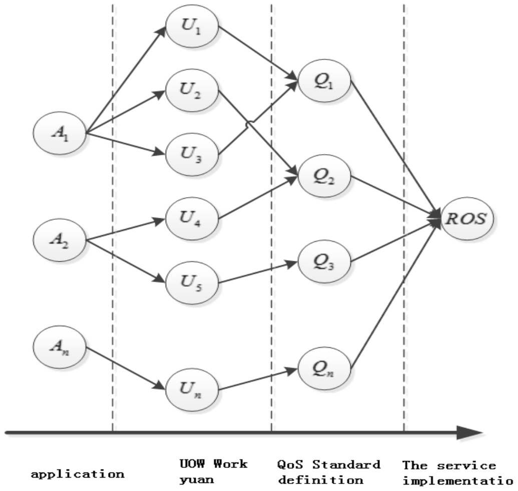

Figure 4. QoS requirements mapping process

The process of extracting the GoW from the application and classifying it by QoS is as follows:

$$NCSApp_i \rightarrow \{G_1^i, G_2^i, ..., G_n^i\} \rightarrow \{Q_e, Q_f, ...Q_j\} \subset \{Q_1, Q_2 ... Q_n\}$$

Where, represents a network control application, represents a GoW, and represents a QoS feature. When GoW classified according to the QoS characteristics, QoS requirements parameters, can be extracted from an application, the user can then according to the QoS parameters and other partners to negotiate, together to make an application to determine the level of QoS based on complete. The specific scheduling of resource services is performed in the middle tier, so the process can be seen as an implementation from the application layer business abstraction to the mid-tier QoS mapping.

C. *QoS component access algorithm*

The problem of component access can be described as: the function configuration diagram for the composite component request is mapped to the corresponding component instance; (2) according to the service QoS requests and system resource condition, choosing component instance specific QoS model, satisfied, and the equation between adjacent component instance can be set up, and all the component instance resources required to be met. In the following figure, an example of QoS component function configuration diagram and QoS diagram is given, and the process of the component instance access algorithm is given.

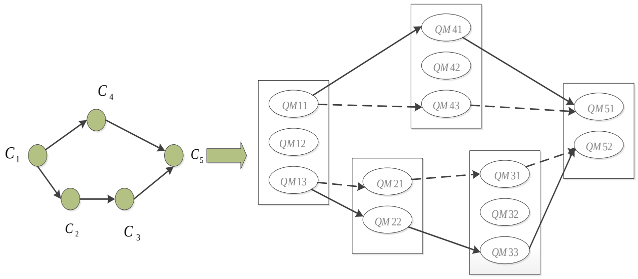

Figure 5. The service component combination process

Service composition for component steps, you will first need to choose for component instance QOS model satisfy the relationship is formed between adjacent component instance, unit construction planning according to the component function configuration diagram CG and QOS of component specifications constructing service QOS diagram QG. The specific algorithm flow is as follows:

Algorithm 1: heuristic component access algorithm.

Input: component function configuration diagram and system current available resource collection.

Output: optional QoS component configuration set -- ConfigSet[]

a) The Edge[N] array is placed in each Edge of the initialization function configuration diagram (CG).

b) Find all source nodes and target nodes in Edge[N] and place them in the array of StartPt[] and EndPt[];

```
c) Foreach u in StartPt
d) $Visited[U] = false$;
e) Foreach W in EndPt
f) $Visited[w] = false$;
g) $ConfigSet = \varnothing$;
h) Foreach u in StartPt
```

```
$QC_{tmp} = \varnothing$;
If $Visited[U] = false$
then Call $Construct(u, QC_{tmp})$;
Procedure $Construct(u, QC)$
{
$Visited[U] = true$;
foreach $QM_i \in QoSProfile[u]${
$QC_{item} = \varnothing$;
    foreach $q_t \in QM_i${
    if( All_Satisfied ==true ) {
            q_Satisfied = flase;
                    foreach $QM_j \in QoSProfile[v]${
                if (Satisfied( $q_t$ )==true)
                    $QC\_Item = QC\_Item \cup \{QM_j\}$;
                        q_Satisfied =true;
                }
        }
  }
if ($u \in StartPt[]$){
        $QC = QC \cup QC\_Item$;}
else if($u \notin StartPt$){
        foreach $Item \in QC${
        if($QM_i \in Item$)
                $QC = QC \cup QC\_Item$;
        }
}
        foreach $v \in EntPt[]${
        $Construct(v, QC)$;
}
}
```

Firstly, the function configuration diagram of the component is initialized, and the source and target nodes in the graph are placed into the corresponding data structure. Set the temporary edge set to empty; Then the heuristics generate configuration set function for each node of the source node; In this function, for each QoS configuration item, check whether the QoS attribute is satisfied in the instance of each component, and if it is satisfied, then check whether the component instance of the destination node satisfies the QoS configuration item; And the recursive method is used until each feasible QoS configuration item is output and the result is output.

IV. SIMULATION EXPERIMENT

In this section, the simulation experiment is used to verify the core algorithm of the QoS management framework and related hierarchical strategy design.

Definition 1: QoS Requirements meeting Rate (QRMR: QoS Requirements Meet Rate), assuming that in all service application requests, the service application is successful, the QoS demand satisfaction Rate is defined as:

$$QRMR = \frac{Success_{request}}{Total_{request}}$$

The simulation experiment of this paper compares QRMR with different component selection algorithms, as shown in fig.6. The simulation experiment environment consists of four PCS networked with QoS component container.

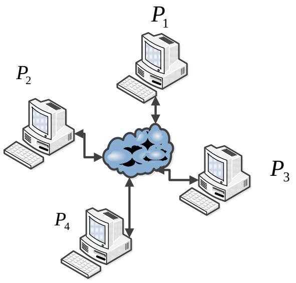

Figure 6. QoS simulation environment network structure

QoS simulation experiment environment hypothesis, influencing QRMR factors for resource constraints and service resource requirements. Simulation experiment, repeatedly randomly from every seconds to create a composite service request, the simulated four composite service function configuration diagram of components and service components deployed as shown in figure 7, and after a 5-1 calculated feasible QOS configuration resource requirements as shown in table 1.

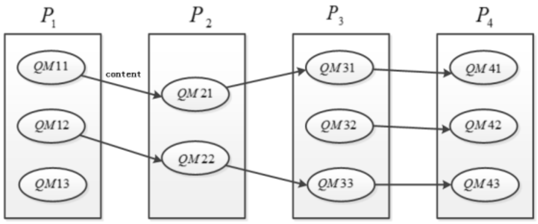

Figure 7. The simulation experiment component function configuration diagram

In the experiment, the QoS component only reserved management for the CPU resources, respectively representing the CPU resources.

TABLE 1. THE COMPOSITE SERVICE QOS CONFIGURATION OF RESOURCE REQUIREMENTS

Service marks	Feasible QoS configuration	$\sum r_1$	$\sum r_2$	$\sum r_3$	$\sum r_4$
S_1	$QM11$	10%	9%	13%	16%
	$QM12$	16%	11%	10%	16%
	$QM13$	9%	15%	6%	4%
S_2	$QM21$	12%	7%	12%	20%
	$QM22$	15%	5%	15%	10%
S_3	$QM31$	7%	15%	4%	9%
	$QM32$	20%	12%	20%	10%
	$QM33$	16%	18%	12%	12%
S_4	$QM41$	10%	8%	8%	7%
	$QM42$	10%	4%	5%	12%
	$QM43$	7%	12%	10%	10%

The QRMR of the statistical system is simulated every 1 second. The duration of each successful service is set in seconds, and when the service is completed, the resource will be recovered by the corresponding QoS component resource management yuan service.In the experiments $Capability_{r1}$, $Capability_{r2}$, $Capability_{r3}$ and $Capability_{r4}$ The size is set to100%.In the component access algorithm, r_1, r_2, r_3, r_4 The weights of resources are set respectively $Q_{r1} = 0.5$, $Q_{r2} = 0.4$, $Q_{r3} = 0.35$, $Q_{r4} = 0.45$.

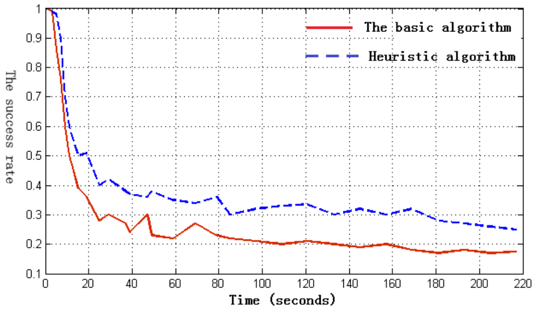

Figure 8. Different components access algorithm's QRMR

Therefore, simulation negotiation algorithm is obtained by using the exhaustive method of integer programming optimal solution. Figure 8 shows the QRMR changed over time. Can see from the result of the experiment at the beginning of the simulation of two kinds of algorithm can achieve higher success rate, but as the simulation process, the heuristic member access algorithm QRMR meet rate is around 0.3, more than the basic component selection algorithm. The results show that the heuristic adjust weight and through service resources through negotiation methods, QRMR can improve.

Can see from the result of the experiment at the beginning of the simulation of two kinds of algorithm can achieve higher success rate, but as the simulation process, the heuristic member access algorithm QRMR meet rate is around 0.3, more than the basic component selection algorithm. The results show that QRMR can be improved by heuristic regulation of resource weight and through service resource negotiation.

V. CONCLUSION

First introduces the basic requirements for networked control systems of QoS, shows an open and scalable QoS semantic model, its constituent elements including contract for QoS parameters, QoS, QoS model, specification of QoS, QoS parameters, etc., makes the service components have the ability to adapted to network resources. Based on the analysis of QoS attribute of service component, the establishment process of component service is described, and a dynamic algorithm of component service group is proposed. At the same time, the design for the application layer, middle layer and resource layer and network layer QoS strategy, finally designs the dynamic service components based on resource distribution of container deployment simulation experiment, this section is verified the proposed QoS goal-driven dynamic deployment effectiveness of the algorithm.

REFERENCES

[1] Xu fei, liu mingyong, li wenbai, lei xiaokang. Study on the digital redesign of robust controllers for parametric uncertainty systems [J]. Systems engineering and electronic technology,2013,01:156-160.

[2] I.T. Foster, C. Kesselman,C. Lee et aL A Distributed Resource ManagementArchitecture that Supports Advance Reservation and Co-Allocation[A]. In: Proceedings of the Seventh International Workshop on Quality of Service[C], London, UK, 1999: 27-36.

[3] Poza-Lujan Jose-Luis, Posadas-Yagüe Juan-Luis.Distributed sensor architecture for intelligent control that supports quality of control and quality of service. Sensors (Switzerland),2015(15),3:4700-4733.

[4] R. J. Al-Ali, K. Amin, G V. Laszewski et al. An OGSA-Based Quality of ServiceFramework [J]. Lecture Notes in Computer Science, 2004, 3033: 529-540.

[5] Hakiri Akram,Berthou Pascal.Supporting SIP-based end-to-end Data Distribution Service QoS in WANs.Journal of Systems and Software,2014(95):100-121.

[6] L. Z. Zeng, B. Benatallah,A. H. H. Ngu et al. QoS-Aware Middleware for Web Services Composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5):311-327.

[7] M. Wieczorek, S. Podlipnig, R. Prodan et aL Bi-criteria Scheduling of ScientificWorkflows for the Gird [A]. In: Proceedings of the 8Th International Symposium onCluster Computing and the Grid (CCGrid 2008) [C], Lyon, France: IEEE ComputerSociety, 2008: 9-16.

[8] Shakkottai S, Rappaport T, Karlsson P. Cross-layer design for wireless networks. IEEE Communications Magazine, 2003, 41(10): 74-80.

[9] A study on the movement and distribution of nodes with multiple entry and exit regions. Computer research and development, 2008, 45(3): 428-435.