Wireless Control Network Model Based on Flexible Time Trigger

Liugang Zheng

Shanghai Monitoring Station the State Radio_Monitoring_Center, Shanghai 201419, China

huner2011@foxmail.com

Keywords: Wireless network, Event trigger, Flexible time trigger, Network model

Abstract. In recent years, with the development of wireless communication technique and urgent requirement of industry, wireless network technology has been widely applied to industry distributed control system. The existing standardized wireless network protocol which has been applied can't meet the above requirements, so there is need to apply some mechanism and design method to establish new and practical wireless distributive real-time system for industry. The paper proposes a wireless control network based on flexible time trigger mechanism to meet the time requirement of real-time application. The paper introduces the concept of flexible time trigger mechanism, and describes wireless control network model based on flexible time trigger mechanism. The paper explains the concept of flexible time trigger mechanism, and expounds how to apply the mechanism to establish wireless control network.

Introduction

Flexible time trigger mechanism is a method designing distributive real-time system. It not only supports time trigger, but also supports event trigger. And it separates two trigger mechanism for time.

Event trigger. In event trigger mechanism, the incidence of a specific event starts the execution of a task. For example, the state of the present system changes, which triggers a task to be executed. Figure 1 describes the opening way of task in an event trigger system. At t1, an external event E1 happens, the incidence of E1 makes task τ1 begin to be executed, and we consider that E1 triggers τ1. The open of τ1 is not certain for time. It opens with the incidence of E1 rather than the coming of t1. So event trigger mechanism has great responsiveness to asynchronous event, but it is not certain for time.

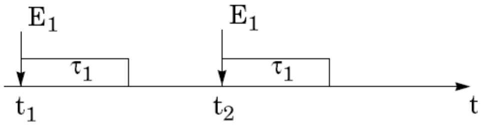


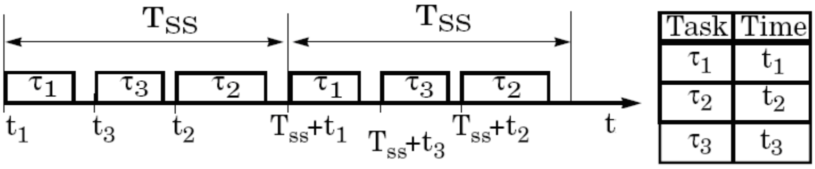
Figure 1 Task scheduling based on event trigger mechanism

In most cases, a processing node operates many tasks. In event trigger system, each task has the event trigger, so there is the situation that there are two events at the same time, which makes two tasks to be started. And there is the situation that two tasks compete CPU. In order to avoid the situation, in event trigger system, all tasks are distributed priority level. When multiple tasks are triggered by the corresponding events, and there is need to use CPU, the task with high priority level achieves the right to use, and the other tasks are in order.

In communication network, the information is obviously not preemptive. When a information is in transmission process, if another node sends information packet, the information crashes. Message collision evidently reduces network utilization efficiency, so it should be avoided as far as possible.

The advantages of event trigger are that it is flexible, it has great responsiveness to asynchronous event, and it can efficiently use system resource. However, applying real-time system of event trigger mechanism is uncertain for time, which makes it difficult for real-time performance evaluation and schedule analysis. In addition, when multiple triggers are triggered at the same time, the task with low priority level may achieve CPU later, which makes the response time go beyond the deadline, which is disastrous for real-time system. When event happens frequently, it can make task switching frequently, which reduces system efficiency.

Time trigger. In time trigger mechanism, the execution of a task is started at specific time. The major component of real-time kernel is clock interrupt, and the main control signal is system clock. In time trigger system, the information required by task scheduling needs to be saves in a dispatch list. The items of dispatch list are the examples of tasks and the opening time. Dispatch list generates by some static scheduling algorithms offline. The algorithms distribute proper opening time for each task instance, which ensures each task instance have its own CPU time slice. So the situations of multiple tasks competing CPU are eliminated. As shown in Figure 2, three cycles of a real-time system are tasks of T. The dispatch list on the right side is generated offline by using static scheduling algorithm. The list includes opening time t1, t_2 and t_3 of task τ_1, τ_2 and τ_3. The basic rule to generate list algorithm ensures that each task is executed before the next task is started. TSS is static scheduling cycle. The kernel executes the same task sequence in each static dispatching period. So the opening time in dispatch list is for the beginning time of static dispatching cycle.



Task	Time
τ_1	t_1
τ_2	t_2
τ_3	t_3

Fig. 2 Task scheduling based on time trigger mechanism

Static scheduling cycle TSS generally takes the lowest common multiple of all task cycles in the system. The case of Figure 2 is very special. The cycle of each task is T, so the static dispatching cycle of the system is TSS=T. The scale of the list of the system is very small, and there are only three items. Each task only can be opened once in a static dispatching cycle. If the cycles of tasks in the system are inconsistent, the scale of the list increases. And some tasks may be started for many times in a static dispatching cycle, which means that a task has many instances in a static dispatching cycle. Obviously, the size of the list is corresponding to the size of static dispatching cycle.

Flexible time trigger system. Time trigger and event trigger has obvious advantages and disadvantages, so both are complementary. And we can combine the advantages of both to implement the system which not only supports time trigger, but also supports event trigger. The system with two mechanisms is flexible and certain, and the system is called flexible time trigger system.

We need to implement the system which not only supports time trigger, but also supports event trigger from two levels, task scheduling level and message scheduling level. Task scheduling is implemented in real-time kernel of mainframe, and message scheduling is implemented in communication controller.

Message scheduling of typical flexible time trigger system is shown in Figure 3. Communication network connects the nodes of multiple processors. And each node has communication controller, processor, I/O interface and local memory. According to the basic cycle structure, the length of network media access is TBC. A basic cycle is divided into static phase ST and dynamic phase DYN. Static phase consists of independent time slice. Only specific static message is distributed in each time slice. The tasks to be sent by static message are started at predetermined time, and each static message relates to specific time slice. In dynamic phase, all nodes can send dynamic message, and the transmission of message are triggered by message instance. Message collision is solved by media access provided by communication network MAC layer.

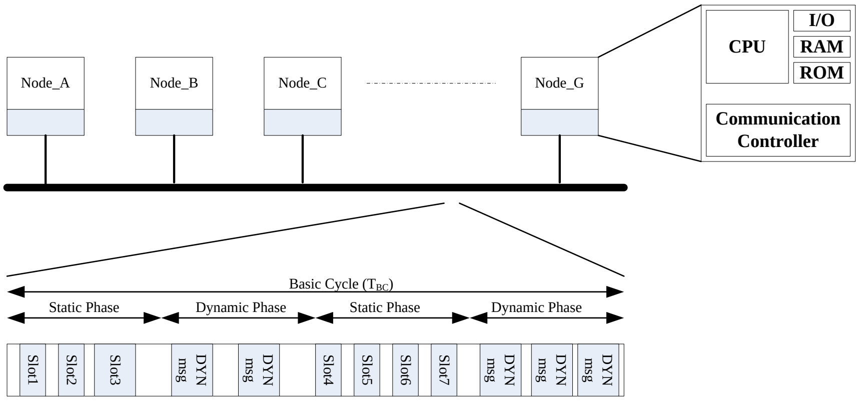


Fig. 3 Flexible time trigger system

Using Flexible Time Trigger Mechanism to Establish Wireless Control Network

Wireless control network model based on flexible time trigger mechanism can be described in Figure 4. There are many wireless nodes in the network, and the model has a hypothesis that all nodes are in the range of single hop, which means that any two nodes in any network can directly communicate without transmitting nodes or routing node.

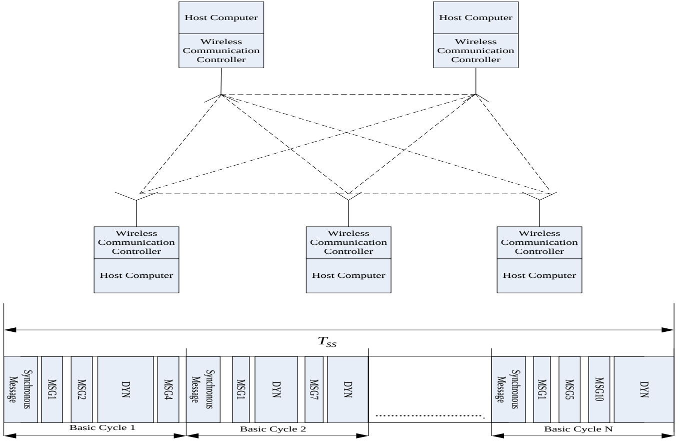


Fig. 4 Wireless control network based on flexible time trigger mechanism

Each node plays different role in the distributive control system. Some nodes are used to collect the data of sensor, and some nodes are used to operate control algorithm, and some nodes are used to drive actuator. But each node has two core components, main frame and wireless communication controller.

The storage of mainframe has dispatching list of event trigger task. Dispatching list of time trigger task specifies that the opening time of all time trigger tasks in a static dispatching cycle. For a specific node, the time trigger task sequences which are started in each static dispatching list is the same.

The storage of wireless communication controller has static message description list. In the static description list, the table entries have sending nodes and receiving nodes of each static message, and the time slice of static message in static dispatching list. In each static dispatching cycle, the static message sequences on the network are the same.

In communication subsystem, a static dispatching cycle is divided into basic cycles. Each basic cycle starts from a synchronous messages, and the synchronous messages are sent by clock host node. And the global clock of the network is based on clock host node. In the paper, the length of basic cycle is represented as TBC.

Synchronous messages tell all nodes that a new basic cycle begins. Basic cycle is divided into time slices, and some time slices are distributed to certain static message. If the idle time slice between two static message time slices is greater than the maximal static message transmission time in the system, the idle time slice can be used as a dynamic phase time slice. For example, in Basic Cycle1 of Figure 8, the time slice in static message MSG2 and MSG4 is dynamic phase time slice.

Message description list not only includes time parameter of static message, but also includes the starting time and length of each dynamic phase time slice, and all nodes can know the time to send dynamic message. All nodes can send message in dynamic phase time slice, so it may cause message competition and message collision. So communication controller should use CSMA/CA mechanism to send information in dynamic phase.

Conclusions

The paper firstly explains event trigger mechanism and time trigger mechanism in details, and compares the advantages and disadvantages. Then, the paper describes the flexible time trigger mechanism which combine the advantages of event trigger mechanism and time trigger mechanism. Lastly, the paper proposes wireless distributive real-time system model based on flexible time trigger mechanism.

References

[1] K. Tindell, J. Clark, "Holistic Schedulability Analysisfor Distributed Hard Real-Time Systems", Microprocessing&Microprogramming, Vol. 50, No. 2-3,1994

[2] E.G. Coffman Jr., R.L. Graham, "Optimal Schedulingfor two Processor Systems", Acta Informatica, 1,1972.

[3] Ganeriwal S, Kumar R, Srivastava M B. Timing-sync protocol for sensor network. In: Proc 1st Int'l Conf on Embedded Networked Sensor System (SenSys 2003), Los Angeles, CA. November 5—7, 2003.138~149.

[4] Traian Pop, Petru Eles, Zebo Peng, " Holistic Scheduling and Analysis of Mixed Time/Event-TriggeredDistributed Embedded Systems " , Proceedings of the10th International Symposium on Hardware/SoftwareCodesign (CODES 2002), Estes Park, Colorado,USA, pages 187-192, 2002.

[5] Liu, C. L. & Layland, J. W. "Scheduling Algorithms for Multi-Programming in a Hard Real-Time Environment." Journal of the Association for Computing Machinery 20, 1 (January 1973): 40-61.