Real-Time Scheduling Algorithm Based on Dynamic Priority for Multi-data Transmission

KONG Zhi-sheng[1] **MA Xiao-yuan**[1] **XU Zheng-yi**[1] **ZHANG Bo**[1] **WEI Jian-ming**[1]

[1]**Shanghai Advanced Research Institute, CAS, Shanghai 201210, China**

Abstract

In response to the demands of heterogeneous data acquisition, data transmission effectiveness and reliability in emergencies as well as environmental wireless monitoring etc., this paper presents a data transmission queue optimization algorithm based on dynamic priority. The algorithm dynamically assign priorities to different data according to their importance, historical transmission data and other factors, as to make the best use of current limited bandwidth of communication to ensure data transmission's real-time degree and effectiveness. Theoretical analysis and simulation show that the algorithm makes quite improvement when applying to data transmission in the environment of non-stable communication link.

Keywords: *dynamic priority task scheduling data transmission queue*

1. Introduction

In the fields of emergency rescuing and environmental monitoring etc., it's required to collect environmental data continuously, and transfer them to a remote receiver. However, because of the amount of data types, wireless transmission link instability and variation of data collection rate, data transmission often suffers from network congestion[1-3]. Therefore, how to ensure reliable and efficient transmission of environmental data under limited wireless channel has been a hot topic.

In emergency rescuing, data types needed to be monitored include meteorological data, poisonous gas concentration data, GPS data, infrared scan data, ECG data etc. These kinds of data differ in lengths, sampling frequencies, refreshing intervals and significance. Also, the unpredictable fluctuations of wireless channel often lead to transmission short-term interruption. Therefore, in order to meet the requirements of timeliness of different data types, data cannot simply be transferred sequentially. There must be a more realistic priority allocation strategy to guide data transmission. In addition, for each type of data, its use contains real-time displaying, as well as analysis and forecasting based on historical data. Both use should be considered in the transmission strategy.

There are several similarities between this problem and traditional task scheduling in real-time systems:

a. Both of them have Contradiction between actual processing ability and required processing ability.

b. Both of them have different tasks. These tasks differ in significance degree, processing time etc.

c. Both of them can rearrange the order of different tasks.

In the conventional scheduling algorithms of real-time systems, different tasks are assigned with different values, execution time, deadlines and other parameters. Meanwhile, multiple tasks compete for system's processing capacity. So, scheduling algorithms aim to allocate time slices and to get as high total value as possible. To achieve this goal, traditional scheduling algorithms gave following methods: first deadline priority, least spare time priority, maximum value priority, maximum value density priority etc.[4-6]. These algorithms are all based on a single perspective. On the basis of them, some researchers bring out hybrid algorithms considering multiple aspects simultaneously [7]. Some researchers also modify some assumptions to obtain improved algorithms[8]. Furthermore, some researchers have pointed out that the value density of tasks' importance during the execution time may not maintain constant, which bringed out the task scheduling algorithm based on dynamic priority[9-11].

Although there are many similarities mentioned above, task scheduling algorithm cannot be simply applied to multi-data transmitting because of the following differences between these two area:

a. In task scheduling model, when beyond the deadline, the importance of a task deceases to 0. While in data transmission, based on different purpose, some data's importance may grow higher when its deadline is exceeded.

b. In task scheduling model, task's overall importance keeps constant, while its density varies over time. While in data transmission, data's overall importance is affected by current time and historical data, it's essentially dynamic.

Based on above analysis, this paper brings out a real-time scheduling algorithm based on dynamic priority for solving data transmission problems in real applications.

2. Mathematics Model

There are 3 core concepts in multi-data transmission under limited wireless channel: data types, data queue and transmission speed.

Definition 1:

Data types set $T = \{T_1, T_2, ..., T_m\}$, each element T_i in this set contains these properties:

s_i : data length of the i-th data type (bytes).

v_i : basic significance value of the i-th data type.

g_i : time interval of the i-th data type (seconds).

$tolerance_i$: the i-th data type's allowable maximum time difference between current time and the timestamp of latest received data of this type (seconds).

r_i: timeout penalty factor of the i-th data type, the rate of contribution value loss when timeout occurs (1/second).

q_i : sliding factor of the i-th data type, with the range of [0, 1]. q_i and $1-q_i$ indicate the proportion of current data type's significance value of as historical data and real-time display.

$mlen_i$: the number limit of the i-th data type in data queue. When exceeded, surplus data should be discarded.

Definition 2:

Data queue $M(t) = \{M_1, M_2, ..., M_n\}$, each element M_i in this set contains these properties:

$type_i$: the type of the i-th data in data queue, element of the set T.

$time_i$: the generation time of the i-th data.

$M(t)$ varies in time. Data transmitted will be removed from this set, newly generated data will be added in this set.

Definition 3:

Transmission speed $BW(t)$ means bytes of data transmitted in 1 seconds, which varies in time.

Moreover, some assumptions should be made because of the need of algorithm design.

Assumption 1:

The contribution value of specific type of data is independent of other data types.

Assumption 2:

Channel is occupied by current transmitting data, until it's finished, when specific data is being transmitted, channel speed keeps unchanged.

Assumption 3:

Other channel consumptions in transmission are ignored.

3. Algorithm

Based on the above mathematical model, this paper designs an dynamic programming algorithm based on priority. This algorithm considers the importance of data types, transmitting speed, allowed time gap, and other factors to assess the potential contribution of current data, as to select out the most suitable transmitting object.

3.1 Algorithm principle

Sender rearrange the original data queue $M(t)$ based on data type $T_1 \sim T_m$ and get multi-queue $Q(t) = \{Q_1, Q_2, ..., Q_m\}$, Q_{ij} is the j-th element in Q_i, its type is T_i, we name its timestamp as $time_{ij}$.

Both sender and receiver maintain a list of update time for each data type: $L(t) = \{last_1, last_2, ..., last_m\}$. $last_i$ is the timestamp of the latest successfully sent/received T_i type data.

In order to determine which one in the current data queue should be chosen to be transferred, each data Q_{ij} should be assigned a priority P_{ij}. There are two factors which affect P_{ij}: the expected value V_{ij} and expected loss D_{ij} of Q_{ij}, each of them is discussed below:

Definition 4:

Expected Value(V_{ij}): Data Q_{ij}'s expected contribution value when it's transmitted to the receiver under current situation of the channel $BW(t)$. V_{ij} is the sum of three components: contribution value as historical data(Va_{ij}), contribution value of maintaining real-time displaying(Vb_{ij}) and contribution value of terminating (if any) timeout punishment(Vc_{ij}). The formula of each of the three parts is shown below:

$$Va_{ij} = v_i q_i \tag{1}$$

$$Vb_{ij} = v_i(1-q_i)u(time_{ij} - last_i) \tag{2}$$

$$Vc_{ij} = r_i(1-q_i)h(time_{ij} - last_i - tolerance_i) \tag{3}$$

$$V_{ij} = Va_{ij} + Vb_{ij} + Vc_{ij} \tag{4}$$

In above formula, $u(x)$ is the unit step function, $h(x)$ is the unit ramp function.

Definition 5:

Expected Loss D_{ij} : When data Q_{ij} is being transmitted under current situation of the channel $BW(t)$, its occupying of the channel will cause other type's data suffering from timeout loss(if any). The formula of calculating D_{ij} is shown below:

$$D_{ij} = \sum_{k=1}^{m} r_k(1-q_k)\,[h(t + s_i / BW(t) - last_k - tolerance_k) - h(t - last_k - tolerance_k)] \tag{5}$$

In above formula, t is current time, $h(x)$ is the unit ramp function.

Definition 6:

Priority P_{ij}: $P_{ij} = V_{ij} - D_{ij}$. By combining formula (1)-(5) we can get the formula:

$$P_{ij} = v_i q_i + v_i(1-q_i)u(time_{ij} - last_i) + r_i(1-q_i)\,h(time_{ij} - last_i - tolerance_i) - \sum_{k=1}^{m} r_k(1-q_k)\,[h(t + s_i / BW(t) - last_k - tolerance_k) - h(t - last_k - tolerance_k)] \tag{6}$$

After calculating each data's priority, choose the data with the highest value as the object to transmit.

To assess the effect of this algorithm, data sender and data receiver both maintain two lists: $V(t) = \{V_1, V_2, \ldots, V_m\}$ and $D(t) = \{D_1, D_2, \ldots, D_m\}$, recording data of each type's contribution value and loss caused by timeout respectively. Defining income as total value minus total loss to compare different transmission scheme.

3.2 Algorithm analysis

Sliding factor q:

Sliding factor q has the range [0,1]. For any type T_i, when $q_i = 0$, Va_{ij} constant equals to 0, while Vb_{ij} and Vc_{ij} grow with $time_{ij}$. That is to say, for queue Q_i, latest data always has the highest priority. After the newest data successfully transmitted, all old data in the queue need not to be saved, for their $time_{ij}$ are less than current $last_i$, which leads to Vb_{ij} and Vc_{ij} both equal to 0, so all old data's V_{ij} equal to 0. As a result, for $q_i = 0$, the queue Q_i only need to save the latest data, which is coincident with the application requirement of "only concern with real-time displaying" shown by $q_i = 0$.

When $q_i = 1$, it's another case. This time $Vb_{ij} = Vc_{ij} = 0$, and for all of the data in the queue Q_i, its Va_{ij} is the same value. Because this type of data has not the need of real-time displaying, each of them has the same value as historical data, so we can transmit them in any order.

For more general case: $0 < q_i < 1$, after similar analysis we can learn that, in the queue Q_i, latest data has the highest priority, and after it's successfully transmitted, remained old data have the same priority, so they can be transmitted in any order until new data arrive in the queue. However, the value of sliding factor q_i determines their priority compared with that of other type's data: the higher q_i is, the higher their priorities are.

Based on the above analysis, for different data types, sliding factor q can control their transmission order.

Algorithm stability analysis:

Another issue to consider is this algorithm's stability, i.e.

whether certain types of data will lead to a situation in which some types of data never get the chance to be transmitted. By examine the formula mentioned above, when a type of data has the demand of real-time displaying, i.e. $q \neq 1$, its Vb_{ij} and Vc_{ij} will grow linearly over time as long as this type of data don't get the chance to be transmitted, while Va_{ij} and D_{ij} won't. For other types of data which get the chance of being transmission, their $last_i$ has been refreshed continuously, so their priorities will be kept within a certain range. The final result is that data with demand for real-time displaying will always get the chance to be transmitted.

Algorithm complexity and control of queue length:

Seen from the above formula of P_{ij}, for data queue with length of n, each time selecting data of transmission, computational complexity is O(n). When the situation of channel is poor, data queue will grow over time. This time, from the perspectives of both the computational complexity or memory consumption, data queue's length must has a limit. For this consideration, each type of data has its queue length upper limit. When data queue of certain type exceeds its limit, oldest data will be dropped.

Treatment strategy for low speed:

In the calculation of this algorithm, transmitting speed $BW(t)$ is used as the divisor to estimate the time required to complete the transmission. When $BW(t)$ equal to or is near to 0, calculation can't be performed or won't give correct result. To handle these situations, algorithm will be suspended when $BW(t)$ is lower than a given threshold until channel situation improves to acceptible status.

4. Data Simulation and Analysis

In order to verify the effectiveness of this mathematical model and scheduling algorithm, based on the actual needs of our project, we conducted a computer simulation with conditions shown below:

Set of data types:

$T_1 : s = 500KB, v = 100, g = 60s, tolerance = 90s,$
$r = 2 / s, q = 0, mlen = 10$

$T_2 : s = 8B, v = 4, g = 5s, tolerance = 10s,$
$r = 0.5 / s, q = 0.5, mlen = 100$

$T_3 : s = 16B, v = 6, g = 3s, tolerance = 10s,$
$r = 0.5 / s, q = 0.3, mlen = 100$

$T_4 : s = 7.5KB, v = 30, g = 10s, tolerance = 30s,$
$r = 2 / s, q = 0, mlen = 100$

The above four data types correspond to infrared scanning pictures, gas concentration data, GPS data, ECG waveform data respectively.

Channel simulation:

Provide four scenarios simulating wireless channel according to whether the channel is stable and/or sufficient.

$BW_1(t)$: Initial transmitting speed is 40kB/s, sustains for 1 minute, then 0kB/s for 10 seconds, and then returns to 30kB/s for 20 seconds, finally stabilized at 10kB/s, sustains for 2 minutes. Total duration is 3 minutes and 30 seconds. This simulates the situation of sufficient but unstable channel.

$BW_2(t)$: Initial transmitting speed is 10kB/s, sustains for 1 minute, then 200B/s for 10 seconds, and then returns to 3kB/s for 20 seconds, finally stabilized at 5kB/s, sustains for 2 minutes. Total duration is 3 minutes and 30 seconds. This simulates the situation of insufficient and unstable channel.

$BW_3(t)$: Initial transmitting speed is 30kB/s, sustains for 2 minutes, then 40kB/s for 1 minute and 30 seconds. Total duration is 3 minutes and 30 seconds. This simulates the situation of sufficient and stable channel.

$BW_4(t)$: 4kB/s for 3 minute and 30 seconds. This simulates the situation of insufficient but stable channel.

Speed threshold: 10B/s.

Simulation process: for each channel situation, data of all types are continuously generated in accordance with their speed. Sender transmits these data twice, first sequentially according to the natural order of the data generated, second according to the priority based on the above algorithm. Data receiver records contribution values and losses of each type of data, and calculates the total income, as to compare the two programs. Simulation results are shown below:

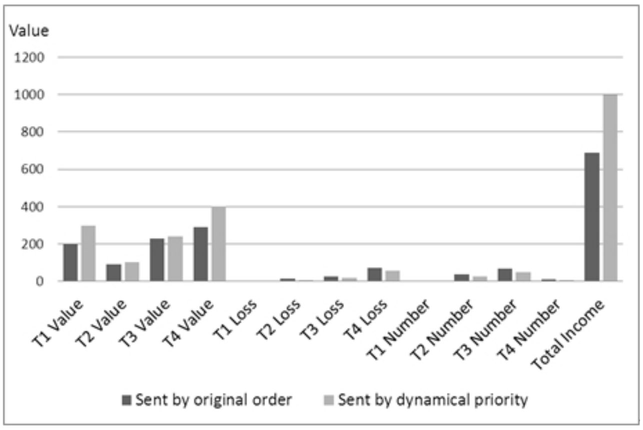

Figure 1: Comparison under sufficient but unstable channel

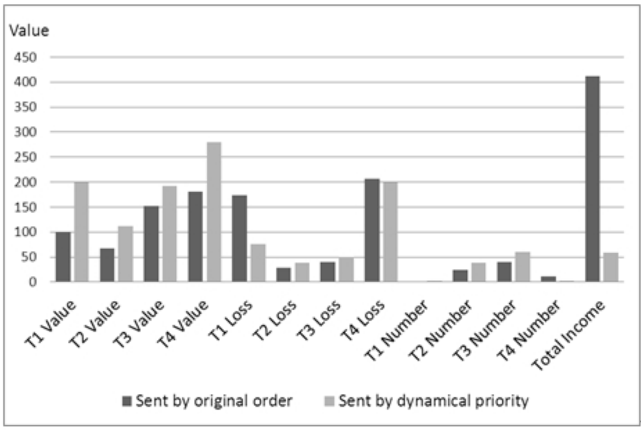

Figure 2: Comparison under insufficient and unstable channel

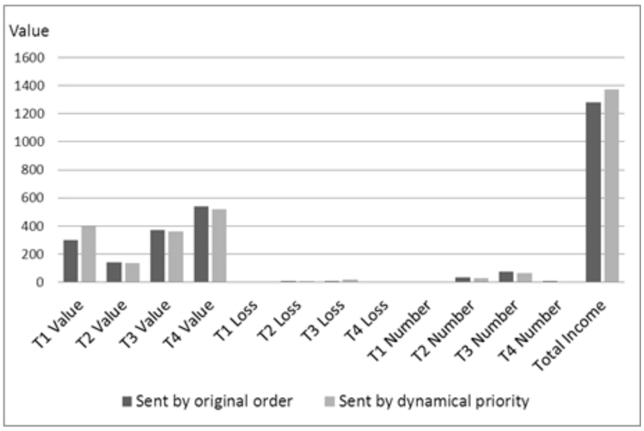

Figure 3: Comparison under sufficient and stable channel

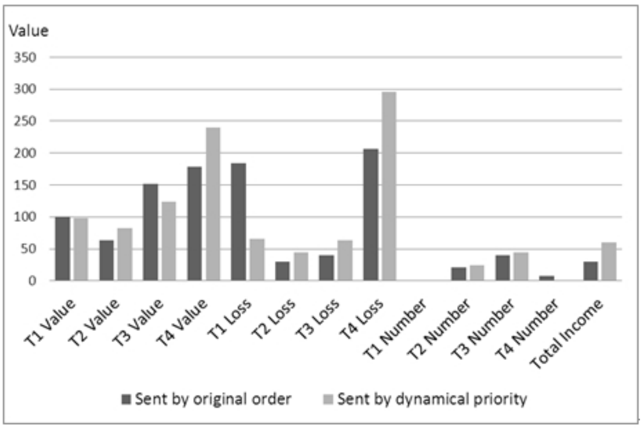

Figure 4: Comparison under insufficient but stable channel

Figure 1 shows the comparison under sufficient but unstable channel. In this situation, even with a short period of disconnection, data will not pile up in the queue. So there is little difference between transmitted data's numbers of each type under two data transmitting schemes. However, dynamic priority algorithm rearranges the order of data transmission, which leads to the growth in contribution values of all the four types of data and significant reduction in the loss of type T_4. As a result, the total income has a substantial growth.

Figure 2 shows the comparison under insufficient and unstable channel. Scheduling algorithm chooses to reduce the transmission of data T_4, in exchange for T_2 and T_3 data’s transmission with higher contribution value. As a result, the total income of scheduled transmission is far more than that of the transmission with natural order.

Figure 3 shows the comparison under sufficient and stable channel. In such case, data won't pile up in data queue, so each data can be transmitted to the receiver without any scheduling. However, as can be seen from the figure, with scheduling algorithm, overall income gets improved slightly.

Figure 4 shows the comparison under insufficient but stable channel. In this situation, data gets transmitted continuously, however, the transmission speed doesn't meet the actual demand, data begins to pile up in the queue. The scheduling algorithm reduces the transmission of data T_4 in exchange for more transmission of data T_2 and T_3. In addition, because of the order's rearrangement, though total transmission amounts of T_1 are the same, the scheduling algorithm has reduced the loss of T_1 significantly. As a result, the overall income of scheduled transmission is better than that of the transmission with natural order.

In summary, under all of the given four different channel situation, this scheduling algorithm can optimize data transmission order and enhance the benefit of the data receiver.

5. Conclusion

This paper analyzes the similarities and differences of real-time system's multi-task scheduling and multi-data trasmitting in unstable wireless channel. Based on task scheduling algorithm, this paper brings out multi-data transmitting real-time scheduling algorithm based on dynamic priority. This algorithm takes data importance, transmitting speed, allowing time gap, historical data transmission situation and other factors into account, estimates potential gains of different types of data to determine their transmission order. Simulation based on actual project data shows that in different situation of channel, this algorithm can bring enhancement of data transmission to fit receiver's demand on data timeliness and effectiveness.

Reference

[1] Zeng Yuanyuan, Xiang kai. Cross-layer Communication Design in Wireless Sensor Networks for Emergency Monitoring Applications. Application Research of Computers, 2012, 29(10):3914-3918.

[2] Zhang Wenbo, Tan Xiaobo. An Improved Bandwidth Allocation Algorithm in Wireless Networks. Journal of Astronautics, 2012, 33(12):1762-1767.

[3] Shuai Xiaoying, Qian Huanyan. Burst Real-time Access Control Protocal Based on PCF. Journal of Nanjing University of Science and Technology, 2012, 36(2):266-271.

[4] Abbott, R.K. & H. Garcia-Molina. Scheduling Real-time Transactions: a Performance Evaluation. ACM Transactions on Database Systems(TODS) . 1992, 17(3):513-560.

[5] Buttazzo, G.C. & M. Spuri & F. Sensini. Value vs. Deadline scheduling in overload conditions. Real-Time Systems Symposium, 1995. Proceedings., 16th IEEE.

[6] Sunita Bansal, Bhavik Kothari, Chittaranjan Hota. Dynamic Task-Scheduling in Grid Computing using Prioritized Round Robin Algorithm. , IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[7] Xu Wenqing, Yang Hongyu. A Dynamic-Priority-Based Rea-l Time Scheduling Algorithm for Mixed Task Set. Journal of Sichuan University (Natural Science Edition), 2006, 43(30:544-548.

[8] Dong Jiwen, Zhang Yang. Improved Task Scheduling Algorithm for Embedded Real-time Operating System and its Application. Journa l of Computer App lications, 2009,29(9):2516-2519.

[9] Dharamendra Chouhan, S. M. Dilip Kumar, Jerry Antony Ajay. A MLFQ Scheduling Technique Using M/M/c Queues for Grid Computing. IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 1, March 2013.

[10] Aldarmi, S.A. & A. Burns. Dynamic Value-density for Scheduling Real-time Systems. Real-Time Systems, 1999. Proceedings of the 11th Euromicro Conference on.

[11] Chen, H. A Dynamic Priority Assignment Strategy for Real-time Task. E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on.