A Round-robin Scheduling Algorithm of Vehicle Gateway Node Based on Self-adaptive Weighted Learning

Xiaoyong Zhang, Jun Peng, Shuo Li
School of Information Science and Engineering
Central South University
Changsha, China
zhangxy@csu.edu.cn

***Abstract*—How to decrease the gateway queue delay in the Train Communication Network (TCN) with heavy communication load is a key problem. This paper models the queue scheduling as a reinforcement learning process and presents a self-adaptive scheduling algorithm of vehicle gateway node based on self-adaptive weighted learning. The proposed algorithm can schedule the queues dynamically. We then compare the simulation results respectively under two circumstance: with enough band resource or with limited band resource. The proposed algorithm's superiority is proved by simulation.**

Keywords-component; gateway queue, self-adaption weight, round -robin scheduling

I. INTRODUCTION

There is always amount of information needs to be transmission in the TCN, and the requirement of real-time is different, so it's very important to control the flow in the gateway nodes.

The classical queue schedule algorithms are First Come First Service (FCFS), Priority queue algorithm and Weighted Round Robin (WRR) [1]. FCFS is simple but can't promise the different requires of QoS; The Priority queue can deal with different kinds of QoS require, but it may lead to too long delays, WRR improve the fairness of the queue schedule but can't provide the upper bound of delay[2].

There are many different types of data converge in the gateway node in TCN, in order to guarantee the real-time of the network and the fairness of queue scheduling, we introduce the idea of differentiated service (diffserv). It need to learn n round robin algorithm that can adjust varies data dynamically, which also can adjust the network resource based on the different priority data flow.

This paper takes advantage of the Reinforcement Learning (RL)[3], model the queue schedule problem as RL process, according to the change of queue's priority and length, adopting the RL algorithm to change the queue scheduling weights, as a result ,we can get the optimal round robin strategy.

II. PROBLEM MODELE

Train gateway can operation complex intelligent scheduling algorithm, so every gateway node can be looked as an agent.

According to the priority level of the gateway nodes' queue from high to low, we define the queues as queue 0, queue 1 and queue 2. Queue 2 adopt the best effort method to transmission data, there is no specific delay require. We acknowledge two defines about the queue:

(1) The packet in data link which arrive at the gateway node should insert an appropriated queue's end based on the require of real-time;

(2) The packet can't be assigned into other queue again once it's had been assigned into one queue;

III. SELF-ADAPTIVE VEHICLE GATEWAY SCHEDULING ALGORITHM BASED ON WEIGHTED LEARN ROUND-ROBIN

Definition 1: State set vector S, which reflects the queue's delay condition.

Given an Scheduled length threshold R_i, The higher of the queue's priority, the smaller of the R_i. At sampling site, collect the length of queue i which is marked as M_i, then compare R_i and M_i. S=$\{s_0, s_1, s_2\}$, Where

$$s_i = \begin{cases} 0 & M_i \le R_i \\ 1 & M_i > R_i \end{cases} .$$

As we all know that, the vector has 8 types value from the best delay {0, 0, 0} to the worst delay {1, 1, 1}.

Definition 2: Action set A. A = $\{a_1, a_2, a_3\}$, where a_i denote the next action to schedule queue i.

Definition 3: Individual return value r_i which can be denoted as (1).

$$r_i = (M_i - R_i)^+ = \begin{cases} 0 & M_i \le R_i \\ M_i - R_i & M_i > R_i \end{cases} \quad (1)$$

r_{sum} is the sum of the return value, $r_{sum} = -\sum_{i=0}^{2} \omega_i r_i$ where $0 < \omega_i \le 1$, i = 0, 1, 2, The larger value of w_i the more sensitive of the queue, meanwhile, the value drive the queue scheduler tend to schedule the queue which in worse delay .

Definition 4: Penalty factor r_H, if scheduling the lower queue when the band resource is shortage, this type of schedule should be punished.

$$r_{Hi} = \begin{cases} 1 & \text{if } R_i \le \min R_j \ \forall j = 0,1,2 \text{ and } j \ne i \\ K & \text{else} \end{cases} \quad (2)$$

where K is an constant and $K > 1$,the total penalty factor r_H can be written as follow:

$$r_H = \sum_{i=0}^{2} \lambda_i r_{H\_i} \tag{3}$$

Where $0 < \lambda_i \le 1$, $i = 0, 1, 2$. Also, the reward function is a scalar of state set to action set mapping .We can judge whether the actions' effect is good or bad, this result can in turn affect the choice of action strategy.

$$r(s,a) = r_{sum} \times r_H \tag{4}$$

Obviously, the largest return value is 0 when all of the queues' delay is meet.

After one scheduling action finishes, the agent can calculate the instant income, then updates the value of Q. The form of Q function is $Q(s, a)$, it expresses that execute action a at state s. In the schedule model of this paper, the Q is a forecast of queue state and return value after scheduling former queue. The larger of Q, the higher probability of the queue being scheduled.

$$\begin{aligned} Q^{\pi}(s,a) &= E[r \mid s_0 = s, a, s'] \\ &= r(s,a) + \gamma \sum P(s' \mid s,a) V(s', \pi) \end{aligned} \tag{5}$$

where s' represents the next state after finished state s. $P(s' \mid s,a)$ is the probability of transfer the state s' after action a. $V(s', \pi)$ is the estimate function.

$$V(s', \pi) = \max_{\pi} Q^{\pi}(s,a) \tag{6}$$

It represents the estimate of the retribution after taking the strategy π. $V(s', \pi) = \max_{\pi} Q^{\pi}(s,a)$.

The learning strategy action is approaching to the queue whose length is large than the schedule queue length R_i. The expression of strategy π is $\pi^*(s) = \arg\max_a Q^*(s,a)$.

Taking the single step Q learning, the update function of Q is in (7).

$$\begin{aligned} Q(s,a) \leftarrow (1-\alpha) Q(s,a) + \\ \alpha[r(s,a) + \gamma \max Q(s', a)] \end{aligned} \tag{7}$$

where α is the learning rate, it's value is $0 \le \alpha \le 1$, γ is the discount factor, it represents the effect of the next Q on the present Q. it's value is $0 \le \gamma \le 1$.Every learning step is greedy, and the goal is $r(s,a) + \gamma \max Q(s', a)$.

Comparing M_i and R_i, we get the state set S of the model, S is changing along with the schedule action. Combine the r and S, the gateway calculates and updates Q, then updates the queue weight according to the value of Q, thus we obtain the queue label A, which is passed on to the gateway scheduler for the queue scheduling.

IV. VEHILCE GATEWAY SCHEDULE

A. Single vehicle gateway schedule

Let Q be the weight of queue schedule, the vehicle gateway adaptive weight round-robin schedule algorithm is shown in algorithm 1.

Algorithm1： Single vehicle gateway adaptive weight Round-robin schedule algorithm

input： R_i, M_i

output： The queue label i which is ready to be scheduled

BEGIN

Initialize Q value table $Q(s,a)$ arbitrarily;

Initialize action set $A = \{a_1, a_2, a_3\}$;

Initialize state set $S = \{s_0, s_1, s_2\}$;

Repeat (for each episode)

Calculate s_i according to R_i and M_i;

Choose a_i from A using policy derived from Q;

Take action a_i, according to $\varepsilon - greedy$;

Make operating system scheduler schedules queue i;

Observe revenue r_i, state s_i';

Update Q value by (8)

$$\begin{aligned} Q(s,a) \leftarrow (1-\alpha) Q(s,a) \\ + \alpha[r(s,a) + \gamma \max Q(s', a)] \end{aligned} \tag{8}$$

Update state set S;

Store Q value in value table;

Until S is terminal.

END

B. Multiple vehicle gateway queue schedule

The communication system in the train is always composed of ten or more vehicle gateways. The vehicle gateway can be looked as agent. Multiple agents enforcement learning is expanded from single agent enforcement learning. Every agent can choose the best strategy based on its own transform state and return value. In order to improve the learning efficiency, we assume that every agent only consider its own action strategy, others are just a part of environment [4].

This paper adopts Coordination graphs[5], the global function Q is divided into local function Q^i. Every Q^i is based on the subset a of global action set.

$$a \in A^i \times A_{j \in \Gamma(i)}{}^j \tag{9}$$

a is the union set of the gateway i's action set and the other gateway's action set which can affect i.

This divide process can be donated by digraph $G = (V, E)$, V is the set of node, which is gateway agent; E is the set of side (i, j), which representations the direct relationship between gateway i and j.

Let S denote all of state; A^i denotes the action set of agent i; P^i: $S \times A^i \rightarrow \Delta(S)$ denotes the state transition function, and $\Delta(S)$ is the probability distribution; r^i: $S \times A^i \times S \rightarrow R$ denotes the return value of agent i and $r^i(s, a^i, s')$ is the return value after agent i finished action a^i from state s to s'; π^i denotes the local strategy of one agent's schedule, so we have π^i : S $\rightarrow A^i$.

$$V^i(s,\pi^i) = E(\pi^i)[r^i \mid s_0 = s] = E(\pi^i)[\sum_{t=0} \gamma^t r^i_{t+1} \mid s_0 = s] \quad (10)$$

Equation(10) is the discount estimation function, r^i_{t+1} is agent *i*'s discount return at time simple t, γ is discount factor, we have $0 \le \alpha \le 1$.

The expression of function Q^i is as follows:

$$Q^i(s,a,s') = \sum r^i(s,a^i,s') + \gamma \sum P(s' \mid s,a^i) V^i(s,\pi^i) \quad (11)$$

For simplicity, we suppose that every agent's action influences its neighbour only. Every agent has one local function

$Q^i(s,a)$, this function value is based on the gateway individual action, but we must take its neighbour gateway into consider.

Q is changeable along with the queue state and its neighbour agent. Considering the neighbour gateway agent, we introduce weigh function

$f(i,j)$, it represents the contribution of agent j due to agent *i* updates its Q. We can formulate process as follows.

$$Q^i(s,a^i) \leftarrow (1-\alpha) Q^i(s,a^i) + \alpha[R + \gamma \sum_{j \in \{i \cup \Gamma(j)\}} f(i,j) \max Q^j(s',a^j)] \quad (12)$$

Under the single vehicle gateway schedule algorithm, we propose the multiple vehicle gateway adaptive weight learning round-robin schedule algorithm.

Algorithm2：Multiple vehicle gateway adaptive weight learning round-robin schedule

input：$R_i, M_i, f(i,j)$

output：The queue label *i* which is ready to be scheduled

BEGIN

Initialize each gateway's Q value table $Q^i(s,a)$ arbitrarily;

Initialize action set $A = \{a_1, a_2, a_3\}$;

Initialize state set $S = \{s_0, s_1, s_2, ..., s_n\}$;

Repeat (for each episode)

Calculate s_i according to R_i and M_i;

Choose a_i from A using policy derived from Q^i in each gateway;

Take action a_i, according to $\varepsilon - greedy$;

Make operating system scheduler schedules queue *i*;

Observe revenue each gateway's revenue r_i and state s'_i;

Update each gateway's Q value by (13)

$$Q^i(s,a^i) \leftarrow (1-\alpha) Q^i(s,a^i) + \alpha[R + \gamma \sum_{j \in \{i \cup \Gamma(j)\}} f(i,j) \max Q^j(s',a^j)] \quad (13)$$

Update state set S;

Store Q value in value table;

Until S is terminal.

END

V. SIMULATION

We simulated the result by Matlab. The simulation topological graph is in Fig. 1.

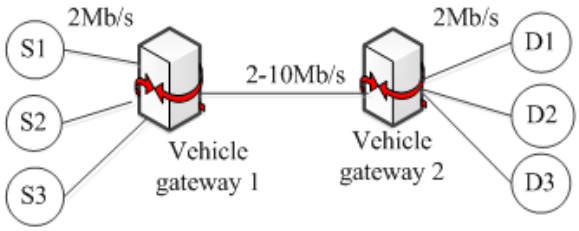

Figure 1. The topological graph of simulation

S1 S2 S3 are the service sending ends and D1 D2 D3 are the receive ends. The simulate time is 60 seconds.S1 is in the highest priority. The priority of gateway queue 1,2and 3 is decreasing by degrees .The R_i, which is the length of scheduled queue 1, 2 and 3 is 200,400,600bits, respectively. We assumed that 50bits is one unit in R_i. The learning rate is 0.7, and the discount factor is 0.98.

1. The compare of queue length when the band resource is sufficient.

The link band is 10Mb/s, the result of queue length compare under the proposed algorithm and WRR algorithm is in Fig. 2.

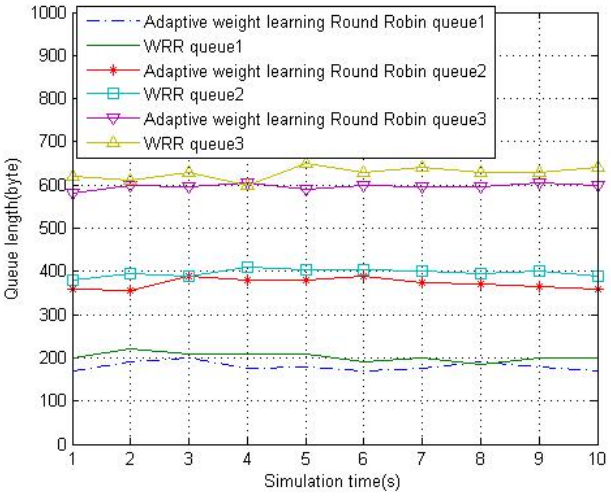

Figure 2. The length of queue when the band resource is sufficient

Obviously, we can study from the result of the simulate .At the beginning of the simulating, the proposed algorithm needed to search the optimal solution. After 2 seconds, the length of queue stayed nearby the R_i gradually.

2. The compare of queue length when the band resource is insufficient.

The link band is 2Mb/s, the result of queue length compare under the proposed algorithm and WRR algorithm is as follows:

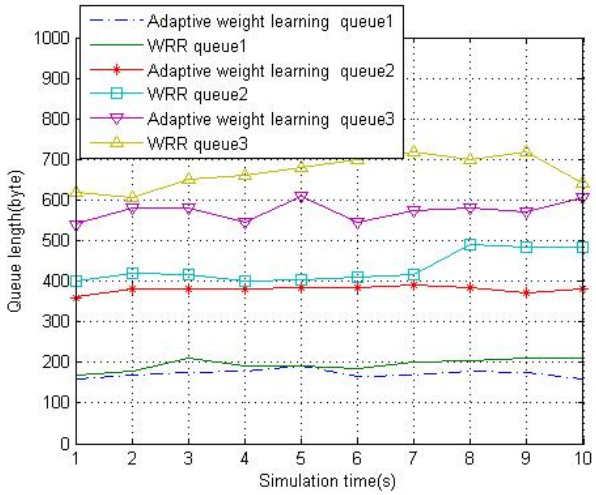

Figure 3. The length of queue when the band resource is insufficient

From the simulate result, we know that both the two algorithm can satisfy the delay requirement, and the length of queue 1 stayed nearby 200bits. The delay requirement of queue 3 also can be achieved in every round, so the algorithm is fairness.

From the simulate result, the proposed adaptive fair learning schedule algorithm can guarantee the real-time transmission of the higher priority, on the other hand, the algorithm shows the superiority on the lower priority data, hence this avoid the case that lower priority data cannot be served.

VI. CONCLUSION

In this paper, we presented the vehicle gateway adaptive weight learning round-robin schedule algorithm. By introducing the agent enforcement learning method, we developed the adaptive weight learning schedule model. Studying the single and multiple vehicle gateway queue schedule algorithm, and then we simulate respectively. The simulate results show that, the proposed algorithm has improvement in the delay and fairness of queue schedule.

REFERENCES

[1] T. Velmurugan, H. Chandra, S. Balaji, “Comparison of Queuing Disciplines for Differentiated Services Using OPNET”, Proc. International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 09), IEEE Press, Oct. 2009, pp.744-746, doi: 10.1109/ARTCom.2009.128.

[2] Y. Zhang, G. Peter, "Performance of a Priority-Weighted Round Robin Mechanism for Differentiated Service Networks," Proc. International Conference on Computer Communications and Networks (ICCCN 2007), IEEE Press, Aug. 2007, pp.1198-1203, doi: 10.1109/ICCCN.2007.4317983.

[3] R. Sutton, A. Barto, Reinforcement learning: an introduction. USA: MIT Press, 1998.

[4] M. Bourenane, “Inductive QoS packet scheduling for adaptive dynamic networks”, Proc. IEEE International Conference on Communications (ICC 08), IEEE Press, May. 2008, pp: 3090-3094, doi: 10.1109/ICC.2008.581.

[5] C. Young-Cheol, “A Survey on Multi-agent Reinforcement Learning: Coordination Problems”, Proc. International Conference on Mechatronics and Embedded Systems and Applications (MESA 10), IEEE Press, July. 2010, pp: 81-86, doi: 10.1109/MESA. 2010.5552089.