Proceedings of the 2005 IEEE
International Conference on Information Acquisition
June 27 - July 3, 2005, Hong Kong and Macau, China

Reinforcement Learning Cooperative Congestion Control for Multimedia Networks

Kao-Shing Hwang, Cheng-Shong Wu, and Hui-Kai Su
Department of Electrical Engineering
National Chung Cheng University
160, Ming-Hsiung, Chia-Yi 621,Taiwan, China
hwang@ccu.edu.tw

Shun-Wen Tan and Ming-Chang Hsiao
Department of Electronic Engineering
We-Feng Institute of Technology
160, Ming-Hsiung, Chia-Yi 621, Taiwan, China
swtan@mail.wfc.edu.tw

***Abstract* - A Cooperative Congestion Control based on the learning approach to solve congestion control problems on multimedia networks is presented. The proposed controller, which is capable of rate-based predictive control, consists of two sub-systems: a long-term policy critic and a short-term rate-adaptor. Each controller in a chained network jointly learns the control policy by real-time interactions without prior knowledge of a network model. Furthermore, a cooperative fuzzy reward evaluator provides cooperative reinforcement signals based on game theory to train controllers to adapt to dynamic network environment. The well-trained controllers can take correct actions adaptively to regulate source flow to simultaneously meet the requirements of high link utilization, low packet loss rate (PLR) and end-to-end delay. Simulation results show that the proposed approach is very effective in controlling congestion of the multimedia traffic in Internet networks.**

Index Terms - Cooperative Congestion Control, Reinforcement Learning, Game theory.

I. INTRODUCTION

A major principle of Internet congestion control is that it is achieved mainly with end-host algorithms, i.e. TCP window based congestion control. However, many related studies have observed that such end-to-end congestion-control solutions are greatly improved when routers have a network-based congestion control **[1]**. A network-based congestion control avoids excessive situations (buffer overflow, insufficient bandwidth and so forth), that can cause a network to collapse, and allows a diverse set of end-to-end congestion-control policies to co-exist in the network **[2]**.

In general, congestion control mechanisms can be divided into three categories: window-based congestion control, rate-based congestion control, and predictive control **[3]**. However, for voice, video and other data services that demand the controllability of packet transfer delay and data rate, windows-based schemes are no longer suitable, because window-based schemes cannot guarantee a minimum communication rate. Closed-loop Rate-based congestion control involves the ideal of handshaking in the network to increase the network utilization, e.g. ATM (Asynchronous Transfer Mode) ABR traffic management **[4]**, additive increase multiplicative decrease (AIMD) algorithm **[5]** and Newman's backward explicit congestion notification (BECN) **[6]**. The effectiveness of feedback control may be diminished when the propagation delay is large compared with the packet transmission time in high-speed data networks. Recent studies have shown that feedback control exhibits oscillations of queue length at a bottleneck node. Predictive control mechanisms can be approached through measuring, statistical and predictive methods, e.g. CSFQ **[7]** and predictive flow control scheme **[8]**. So far, predictive control may be a good solution, but the accuracy of these statistical and predictive methods affects the eventual performance directly.

In this paper, we study explicit data rate (ER) based congestion control strategies based on a cooperative learning approach in the backbone networks, where the learning is performed on the core node cooperatively, and the edge nodes are commanded by the core nodes to regulate the incoming traffic. Since the transmission speeds of data traffic on the core nodes in the backbone networks are very high that all computationally intensive traffic regulation functions such as frame discarding or varying cording rate in video transmission are pushed to the edge node in our study. Based on these concepts, we propose an adaptive congestion control method, called Cooperative Congestion Controller (CCC), to solve this problem. CCC has the characteristic of rate-based control and predictive control. In particular, it can control the applicable flow rates immediately and its efficiency is unaffected by the propagation delay and network scale. Reinforcement learning with temporal difference (RL/TD) **[9]** methods adopted in CCC can learn empirically without prior information about the environmental dynamics. It has two artificial neural networks (ANNs) to learn to achieve cooperative Nash equilibria **[10]**. The first network is the policy-learning model, whose purpose is to select stochastic actions corresponding to estimation of the second part that is an action-value predictor **[11]**. Thus, each part needs an individual set of eligibility traces to enhance the convergence speed; i.e., one for each state and one for each state-action pair.

II. COOPERATIVE CONGESTION CONTROL FRAMEWORK

The goal of CCC is to not only avoid network collapse and improve network QoS, but also to make the network utilization as high as possible. Low packet delay and packet loss of the network QoS to a specified users or services are supported in a CCC network domain. The framework has three key aspects. First, to avoid per-flow buffering and scheduling, we use a simple FIFO queue in each router. In addition to fast forwarding and routing, every core router only has to monitor the state of its FIFO queue. Second, each core

0-7803-9303-1/05/$20.00 ©2005 IEEE

router is performing the CCC scheme and learning the network congestion status with other core routers. Instead of conventional feedback mechanisms and complex statistical methods, every core route can acquire the network situation rapidly and exactly. Third, each edge router or Internet gateway implements a rate-based load control scheme so the Internet gateway will be notified the network status by the adjacent core routers. Also, it will control the transmission rate of incoming traffic directly if the adjacent core router receives the network situation. Since the incoming traffic is effectively controlled well in the Internet gateway, this CCC scheme can improve the crisis condition.

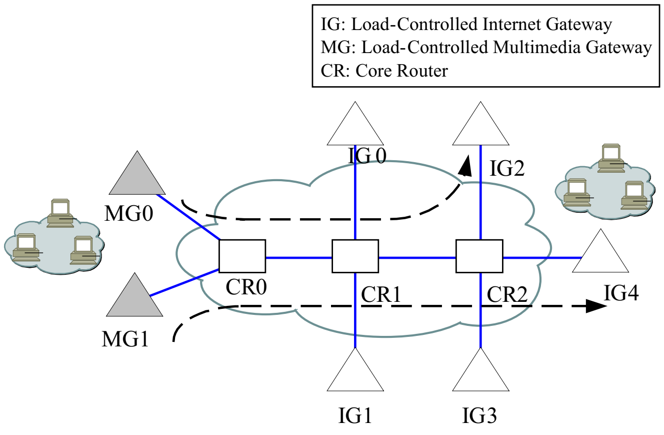
IG: Load-Controlled Internet Gateway
MG: Load-Controlled Multimedia Gateway
CR: Core Router
IG 0
IG2
MG0
CR0
CR1
CR2
IG4
MG1
IG1
IG3

Fig. 1 A topology of simulated network

A. System Configuration

As shown in Fig. 1, there are three components in the network domain, including load-controlled Internet gateway (IG), load-controlled multimedia gateway (MG) and core router (CR). IGs are edge routers, which are connected to other network domains or Internet users, and they also support traffic control. If the Internet traffic is to be delivered from an ingress IG to another egress IG through the core network, the traffic will be aggregated into a virtual path or circuit in the ingress IG and delivered over the path to the egress IG. Finally, that will be delivered to each end host from the egress IG. If the network is congested, the transmission rate of the path will be reduced in the ingress IG. Otherwise, it will be transmitted as much as possible to make best use of the available bandwidth.

MGs not only support traffic control, but also provide dynamic-codec rate control for a video or audio stream, which can dynamically select a suitable code rate to change the transmission rate. On the other hand, the transmission rate of the multimedia stream may be decreased by omitting the nonsignificant frames or bits, while the network is congested. That is, in our scheme nonsignificant frames or bits of video streams in the same virtual path are discarded first on the MG rather than dropping packets of significant frames or bits on the core router due to network congestion. Thus, the multimedia service is not broken off while changing the coding rate and only some sensitive users may feel the quality of video is degraded.

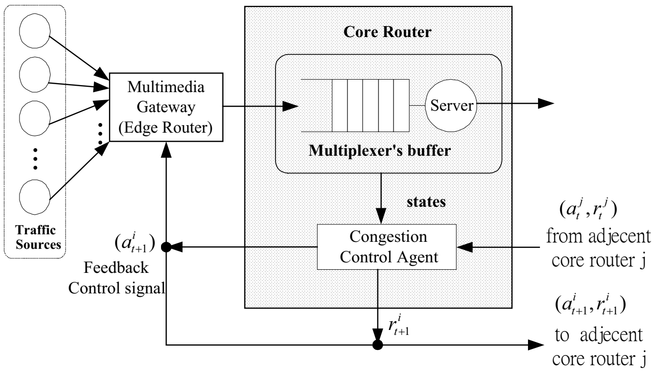

Fig. 2 The framework of the feedback congestion control system.

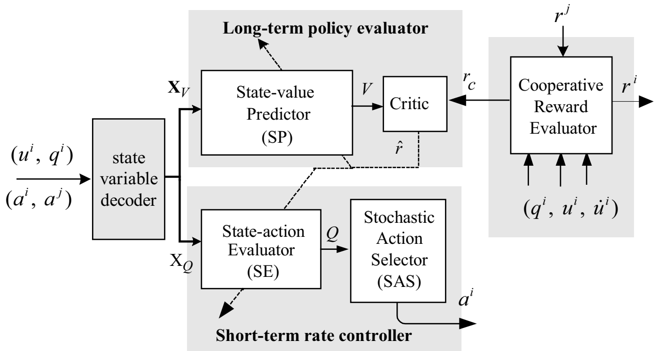
Long-term policy evaluator
X_V
State-value Predictor (SP)
V
Critic
r_c
$\hat{r}$
r^j
Cooperative Reward Evaluator
r^i
$(q^i, u^i, \dot{u}^i)$
(u^i, q^i)
(a^i, a^i)
state variable decoder
X_Q
State-action Evaluator (SE)
Q
Stochastic Action Selector (SAS)
a^i
Short-term rate controller

Fig. 3 The schematic structure of CCC for node *i*; bold lines indicate the path of signals and thinner lines represent the path of a single scalar.

CRs are located in the network center of the congestion control domain and they are not directly connected with other network domains or users. Because they deliver heavy traffic along the backbone, the action of CRs must be simple and effective. In addition to fast forwarding, CRs also need to track the status of the FIFO queue and perform the CCC scheme. They exchange the rewards of CCC learning model with other CRs and control the adjacent IGs and MGs to adjust the adaptable transmission rate of the paths. Since, the transmission rate can be regulated on the IGs and the MGs, and the packet loss in the core network is minimized. In addition, if the bandwidth is sufficient and available in the backbone, the traffic will be transmitted with the maximal rate. Therefore, the network utilization is not sacrificed due to the CCC scheme.

B. Architecture of CCC

In order to reduce the complexity of the cooperative control, the CCC rewards are exchanged only along a virtual path to guarantee its QoS. The virtual paths arriving multimedia traffic streams can be tuned to the adaptive transmission rate without interrupting video service because of dynamic coding rate controlled on MGs. The virtual paths carrying Internet traffic also can be forced to adapt this network condition on IGs. Due to end-to-end TCP scheme in the Internet hosts, the sending rate of Internet traffic aggregated in the virtual paths is slowed down or speeded up automatically. CCC can be easily adapted to conventional or advanced high-speed networks, such as ATM (Asynchronous Transfer Mode) or MPLS (Multi-Protocol Label Switching) networks. The CCC rewards can be carried in an OAM (Operations, Administration and Management) packet or an OAM cell, which is used for exchanging OAM information along a virtual path or circuit in an MPLS or ATM network. Thus, it is not necessary to design a new control protocol for the CCC scheme.

TABLE I
THE SPECIFICATIONS OF THE VARIABLES OF CCC FOR NODE *i*.

q	queue lengths in a core node
u	explicit sending rate of sources
s_t	state vector delineated by variables q and u at time t
V	output of SP network
V'	expected evaluation value of state vector s_t
Q	output of SE network
a_t^i	feedback control signal generated by the SAS element
r^j	reinforcement scalar signal provided by neighbouring node j
r^i	reinforcement signal generated by node i
r_c^i	Cooperative reward $r_c^i = \mu^i r^i(s_t) + \mu^j r^j(s_{t-1})$
$\hat{r}$	heuristic reinforcement signal $= r_c^i + \gamma V^i(s') - V^i(s)$

The proposed framework of CCC involves multiple agents and mechanisms for coordinating the behaviour of individual agent to guarantee the quality of services (QoS) in multimedia networks. The design and analysis of a situation with multiple interacting agents to congestion control problems are needed to consider the common goals of these agents, their possible actions, and the information available to each agent. Each CCC in core nodes learns cooperatively according to the status of its own queue lengths and data rates, along with the sending rates and rewards provided by the neighbouring CCCs. Only two CCCs are needed to construct a co-learning pair, i.e., a two-agent stochastic game, so that feasible and efficient learning is possible. In this situation, as more nodes are included in the networks, the complexity of the CCC remains almost unchanged.

Any specified congestion control agent can be located at a core router, as shown in Fig.2. In the AIMD case, the agent senses the system's states and makes a decision based on a rate control scheme to avoid packet losses and increase the utilization of the multiplexer's output bandwidth. The proposed CCC can behave optimally without explicit knowledge of the environment, relying only on the interaction with unknown environment and provide the best action for a given state. Each CCC consists of two subsystems: the state-value predictor (SP) element is the long-term policy selector while the short-term rate controller is composed of the state-action evaluator (SE) element and the stochastic action selector (SAS) element. Topologically, two neural networks jointly implement the structure of the SP and the SE networks. The SE and the SAS networks produce an optimal control signal *a* in response to the status of Internet networks. The CCC persistently receives the state inputs decoded by a state decoder and accordingly performs a congestion avoidance action to affect the states of environment. The specification of the variables of the CCC network is depicted in Table I.

III. ON-LEARNING OF CCC

In high-speed networks all CCCs in core nodes follows the BOXES scheme in quantizing state space and constructing a neural network controller to regulate the sending rate of sources as shown in Fig. 3. The state space is composed of four variables (a^i, a^j, q^i, u^i) : the current action a^i , the current queue length q^i, the current rate of sending rate u^i for agent *i*, and the current action a^j for agent *j*. The state variables are sampled and quantized into two *n*-component binary vectors, one for $\mathbf{X}_V$ and the other for $\mathbf{X}_Q$. The components of each vector are all zeros except for one in the position corresponding to the state of the system at that instant. The set of all possible action is denoted by A^i for agent *i*. In addition, $v[s]$ is the value function predicated for states s, and $Q[s][a^i][a^j]$ is the action-value for state s after action-pair (a^i, a^j) is applied; while $x[s]$ and $e[s][a^i][a^j]$ are the eligibility traces used to update $v[s]$ and $Q[s][a^i][a^j]$, respectively. The partitions of state space are based on the following quantization thresholds:

1) q^i : queue length of core nodes equally divided by ten partitions numbered 0 through 9.

2) a^i , a^j : feedback adopted actions of sources at 0.25, 0.5,0.75,1.0

The controlled sending rate is defined by the equation:

$$u_t = a_t u_0, \tag{1}$$

where $a_t = (0.25,\ 0.5,\ 0.75,\ 1.0)$ is the feedback signal provided by CCC. The controlled sending rate of sources will take the following values according to the feedback control signal: $u_t = u_0,$ $u_t = 0.75\ u_0,$ $u_t = 0.5\ u_0,$ and $u_t = 0.25\ u_0$, where u_0 is the highest output rate of the source at a specified load.

In high-speed networks all CCCs in core nodes adopt the same reward structure to work cooperatively so as to maximize their common expected reward per an action and to avoid all penalties, which might occur at any time. Reinforcement signal r for a specified state is denoted as $r = 1$ for award, and $r = 0$ for penalty. The reward is given, when satisfying one of these three rules:

1) rule 1 : $q_L < q_{t+1} < q_H$ and ($\dot{q}_{t+1} > 0$ or $\dot{u}_{t+1} > 0$)

2) rule 2: $q_{t+1} < q_L$ and ($\dot{u}_{t+1} > 0$ or $u_{t+1} = u_0$)

3) rule 3: $q_{t+1} > q_H$ and ($\dot{u}_{t+1} < 0$ or $u_{t+1} = 0.25\ u_0$)

The cooperative reinforcement signal is calculated as follows:

$$r_c^i = \mu^i r^i(s_t) + \mu^j r^j(s_{t-1}) \quad \text{for CCC } i, \tag{2}$$

where μ^i and $\mu^j \in [0,1]$ are weighting factors; r^i is a reinforcement signal of agent *i*, and r_c^i is its cooperative reinforcement signal; r^j is a reinforcement signal of neighbouring core node *j*.

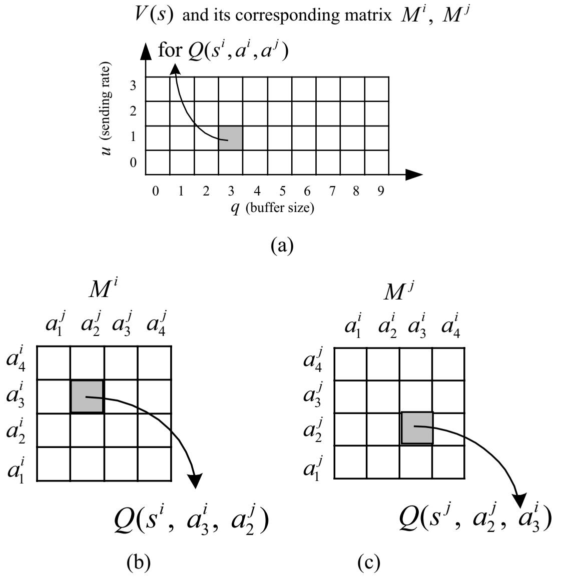

Fig. 4 (a) Input space partitioning for $V(s^i)$ and $Q(s^i, a^i, a^j)$ of node i; its corresponding bi-matrix. (b) for node i and (c) for cooperative node j

Each CCC invokes the learning algorithm to perform the two tasks of the evaluation and action selection, which are implemented by the SP element and the SE element, respectively. For more efficient learning, CCC learns via the TD method with eligibility trace, which adjusts its weights in proportion to the difference between reinforcement predictions on the consecutive steps. Hence, the internal reinforcement signal can be calculated using the temporal difference error as

$$\hat{r}_t = r_{c,\,t+1} + \gamma V_{t+1} - V_t. \tag{3}$$

where $r_{c,\,t+1}$ is a reinforcement signal; $\hat{r}$ is the TD error; and γ is the discount rate.

The following equations illustrate the operation of the SP element to update **v** and $\overline{\mathbf{x}}$ for all states, using:

$$v_{t+1}^k(s) = v_t^k(s) + \beta \hat{r}_t \overline{x}_t(s), \tag{4}$$

$$\overline{x}_t(s) = \begin{cases} \lambda \overline{x}_{t-1}(s) + (1\text{-}\lambda), \text{ if } s_{t+1} = s \\ \lambda \overline{x}_{t-1}(s), \text{ otherwise} \end{cases}, \tag{5}$$

where $v(s)$ is the value function predicted for a state *s*; $\overline{x}(s)$ is the eligibility traces used to update $v(s)$; λ is the trace decay rate; β is the positive learning rate.

On the other hand, at each synapse of the SE element there are two trace signals: the long-term trace $Q_t(s, a^i, a^j)$, which evaluates the performance of the action taken for state *s*; and the short-term trace $e_t(s, a^i, a^j)$, which is required to update the long-term trace. When entering a state, agent *i* will adopt an action base on the action selected by agent *j*, as shown in Fig 4. The rules of SE to update **Q** and **e** for all states are expressed as follows:

$$Q_{t+1}(s, a^i, a^j) = Q_t(s, a^i, a^j) + \alpha \hat{r}_t e_t(s, a^i, a^j), \tag{6}$$

$$e_t(s, a^i, a^j) = \begin{cases} \lambda e_{t-1}(s, a^i, a^j) + (1-\lambda)\cdot a^i, \text{ if } s_{t+1} = s \\ \lambda e_{t-1}(s, a^i, a^j), \text{ otherwise} \end{cases} \tag{7}$$

The SAS element adopts a Softmax action selection method to choose an action with probability, such as

$$\text{Prob}\left\{ a_{t+1} = a^i \mid s_{t+1} = s \right\} = \frac{\exp\left(Q[s][a^i][a^j]\right)/T}{\sum_{a^k \in A^i} \exp\left(Q[s][a^k][a^j]\right)T} \tag{8}$$

where *T* is a temperature parameter for an annealing process.

Experimentally all agents adopt the same values for all CCCs. We obtain these values through trial and error; that is, $\alpha = 0.5$, $\beta = 0.5$, $\lambda = 0.9$, and $\gamma = 0.5$. Complete on-line learning algorithm of an individual CCC is depicted in the following steps:

1. Initialize only for the first time:
   In the SP, for all $s \in S$: $v[s] \leftarrow 0, x[s] \leftarrow 0$.
   In the SE, for all $s \in S$ and $a^i \in A^i$, $a^j \in A^j$:
   $Q[s][a^i][a^j] \leftarrow 0, e[s][a^i][a^j] \leftarrow 0$.
2. For each step *t*+1 of learning:
   Obtain the state of the environment, s_{t+1}.
   Receive the cooperative reward, $r_{c,t+1}$, from the cooperative reward generator.
   Calculate the internal reinforcement signal, $\hat{r}$, using
   $\hat{r} \leftarrow r_{c,\,t+1} + \gamma \times v[s_{t+1}] - v[s_t]$.
3. In the SP, for all $s \in S$:
   Update ***v*** and $\overline{\mathbf{X}}$ with $v[s] \leftarrow v[s] + \beta \times \hat{r} \times \overline{x}[s]$
   and $\overline{x}[s] \leftarrow \begin{cases} \lambda \times \overline{x}[s] + (1-\lambda) \text{ , if } s_{t+1} = s \\ \lambda \times \overline{x}[s], \text{ otherwise} \end{cases}$.
4. In the SE, for all $s \in S$ and $a^i \in A^i$, $a^j \in A^j$:
   Update ***Q*** and ***e*** with
   $Q[s][a^i][a^j] \leftarrow Q[s][a^i][a^j] + \alpha \times \hat{r} \times e[s][a^i][a^j]$
   and

$$e[s][a^i][a^j] \leftarrow \begin{cases} \lambda \times e[s][a^i][a^j] + (1-\lambda)\, a^i \ , & \text{if } s_{t+1} = s \\ & \text{and } a_t = a^i,\ a_{t-1} = a^j \\ \lambda \times e[s][a^i][a^j]\ , & \text{otherwise} \end{cases}$$

5. Select an action a_{t+1}, by uniform random number generator N(0,1), from actions pair $(a^i,\ a^j) \in A$ with probabilities,

$$\text{Prob}\left\{ a_{t+1} = a^i \ \middle|\ s_{t+1} = s \right\} = \frac{\exp\left(Q[s][a^i][a^j]\right)/T}{\sum_{a \in A^i} \exp\left(Q[s][a][a^j]\right)/T}$$

6. Apply the action, a_{t+1}, to the environment, and go to step 2 for next step of on-line learning.

TABLE II
TRAFFIC PATTERN OF OFFERED LOAD.

Source	Offered load (Mbps)	Destination	Packet size (bytes)	Arrival type	Traffic type
MG0	30~60	IG2	1500	Poisson	Type-I
MG1	30~60	IG4	1500	Poisson	Type-I
IG0	40	IG2	64 ~ 1500	Poisson	Type-II
IG1	40	IG4	64 ~ 1500	Poisson	Type-II
IG2	40	IG4	64 ~ 1500	Poisson	Type-II
IG3	40	IG4	64 ~ 1500	Poisson	Type-II

IV. SIMULATION AND COMPARISONS

Comparisons between no control, AIMD, and CCC are analysed by an event-driven program coded in C language, which designed and implemented for simulations on network supporting multimedia services. The simulated network environment consists of two multimedia gateways, three core routers with a finite buffer length of 30, and five load-controlled Internet gateways, as shown in Fig. 1. In the simulations, the output capacity of the transmission links of core routers is 100 Mbps. The input traffic to load-controlled Internet gateways is categorized into two types: video and voice services for Type-I traffic, and Internet services for Type-II. Table II shows the offered traffic pattern, packet lengths, and the relationships of sources and destinations in the simulations. The heavy controlled traffic is characterized by a randomized packet size of 64~1500 bytes and a Poisson arrival rate with mean 40 Mbps. In the simulations, the congested node CR1 aggregated the traffic of MG0, MG1, IG0, and IG1; therefore the maximum traffic is 200 Mbps (offered load = 2.0).

The control scheme of AIMD is precisely implemented by a two-threshold congestion control method. In the CCCs scheme, the controlled sending rate is determined by the equation (1). For the link utilization of the proposed CCC schemes, we focus on the observation of the link utilization at congested node CR1, as shown in Fig. 5. Clearly, the no control method has higher link utilization because the input rate of offered loads to core routers is not regulated. But it is easy to cause high packet losses and long packet delay simultaneously. In contrast, the other control methods can throttle the arrival rate in response to time-varying heavy traffic but still can take the best use of available bandwidth. Because the no control method does not regulate the input rate, a great deal of traffic pouring into the buffer of CR1 can cause its queue size to grow rapidly, implying that high packet loss and a long queuing delay will occur. The mean queue length under various multimedia-offered loads interfered by heavy controlled traffic are shown in Fig. 6. Both schemes CCC and AIMD can regulate input traffic appropriately. In particular, CCC can regulate input traffic accurately by means of cooperate learning between core routers. Therefore, the mean queue length of CCC is lower than that of AIMD. Figure 7 shows the packet loss rate for the outgoing link of MG0, clearly indicating that no control method has higher packet loss rate as the offered load is increasing. The reason is that the externally offered load is larger than the capability of CR1; hence, most packets blocked in the CR1 buffer are waiting for transmission, thus causing high packet loss and long packet delay. In particular, while the offered load to MG0 is 60 Mbps with heavy controlled traffic, the packet loss rate is deteriorated. Even though we adopt the AIMD scheme to regulate the sending rate for responding to the rapidly fluctuating traffic, high packet loss rate still occurs due to its reactive control. However, the CCC can decrease packet loss rate greatly, whether the disturbance load is heavy or not. Figure 8 shows the mean delay of the MG0 link. Figure 9 shows the PMF distribution of feedback control signal of CR1. Core routers transmit a control signal to IGs every 0.1ms periodically to regulate input traffic adaptively according to system dynamics. In light controlled traffic case, CCC in CR1 mostly adopts full speed to transmit packets form sources to various destinations, but only a small portion of low rates are used to accommodate traffic to avoid packet losses. As the traffic loading becomes heavier, conservative policies are preferred at the expense of throughputs, hence the percentage of higher rates decreases and that of the lower rate increases in order to avoid congestion. This shows core routers use various policies by co-learning with adjacent core routers to provide high link utilization, low packet loss rate in complicated network environments.

V. CONCLUSIONS

This paper proposes a game-theory-based cooperative congestion control method for solving congestion control problems on multimedia networks. The CCC controller in a chain network jointly learns the control policy, based on game theory, by real-time interaction without prior knowledge of a network model. Furthermore, a cooperative reward evaluator provides cooperative reinforcement signals to train controllers to adapt to varying network conditions. The well-trained controllers maintain an expectation of reward and take the best action to control source flow. In the simulations observe the QoS (Quality of Service) of a multimedia streaming application in a CCC-based network and in an AIMD-based network. Multimedia streaming applications need much available bandwidth and they are more sensitive to packet delay and packet loss than other data streaming applications. The results show that CCC is more adaptive to real-time applications than the AIMD scheme for the in Internet. Under the constraint of maximum network utilization, CCC can minimize the packet delay and packet loss more effectively than the AIMD scheme.

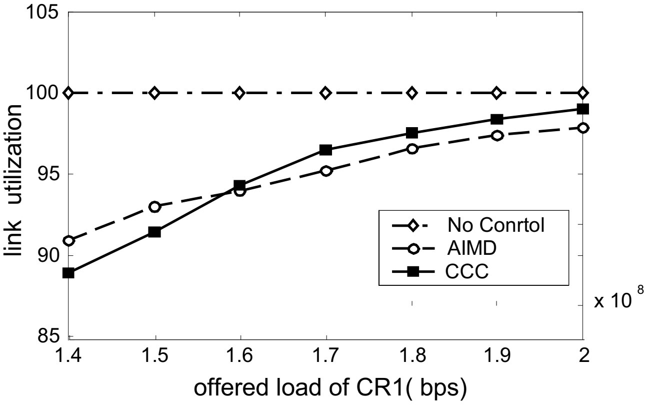

Fig. 5 Link utilization of CR1 under heavy controlled traffic from IG0 and IG1

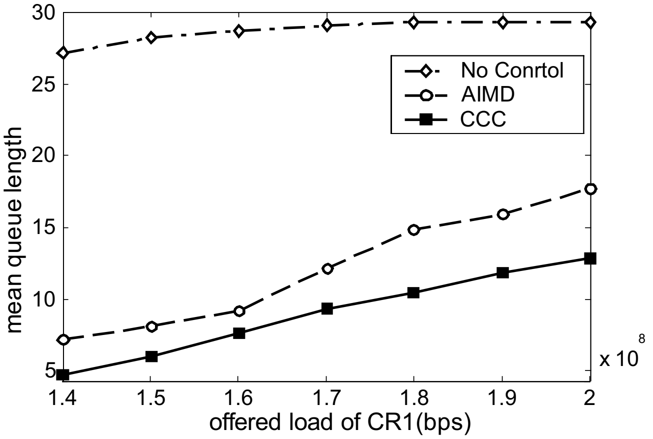

Fig. 6 Mean queue length of CR1 under heavy controlled traffic from IG0 and IG1

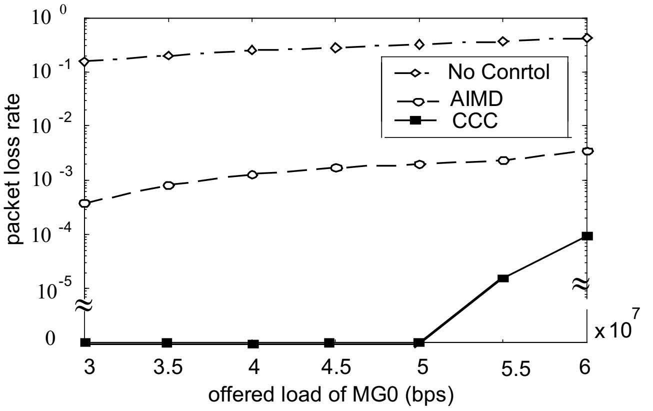

Fig. 7 Packet loss rate of MG0 outgoing link under heavy controlled traffic from IG0 and IG1.

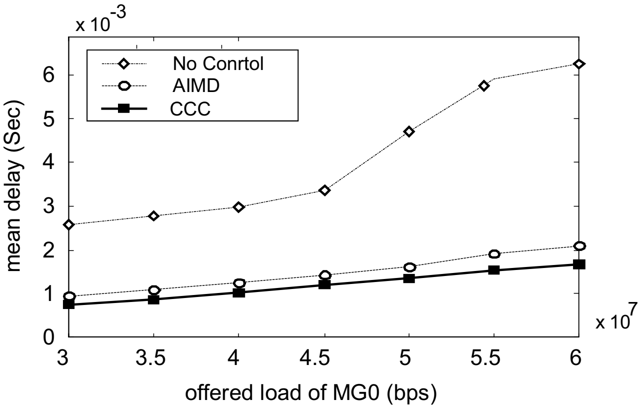

Fig. 8 Mean delay of MG0 link under heavy controlled traffic from IG0 and IG1

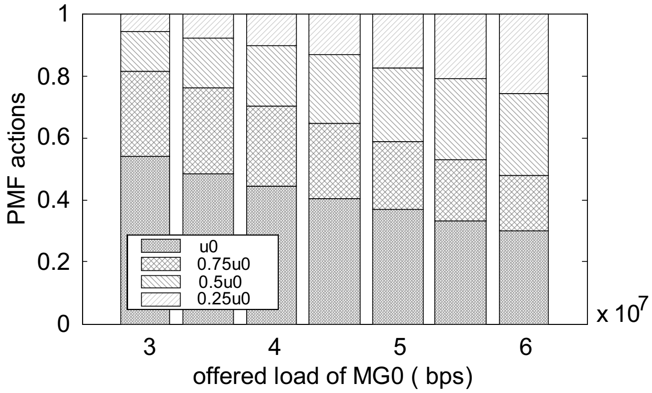

Fig. 9 PMF actions of CR1 under heavy controlled traffic from IG0 and IG1

References

[1] J. Nagle, "On packet switches with infinite storage," IEEE Trans. Commun., vol. 35, pp. 435-438, Apr. 1987.

[2] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," J. Internetw. Res. Experience, pp. 3-26, Oct. 1990.

[3] H. J. Chao and X. Guo, *Quality of Service Control in High-Speed Networks*, John Willey & Sons, 2002.

[4] ATM Forum, *Traffic Management Specification*, Version 4.1, AF-TM-012.000, Mar. 1999.

[5] D. M. Chiu and R. Jain. "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," Computer Networks and ISDN Systems, vol. 17, pp. 1-14, 1989.

[6] P. Newman, "Traffic management for ATM local area network," IEEE Commun. Mag., vol. 32, no. 8, pp. 45-50, Aug. 1994.

[7] I. Stoica, S. Shenker and H. Zhang, "Core-Stateless Fair Queueing: A Scalable Architecture to Approximate Fair Bandwidth Allocations in High-Speed Networks", IEEE/ACM Trans. Netw., vol.11, no. 1, pp. 34-36, Feb. 2003.

[8] D. Qiu and N. B. Shroff, "A predictive flow control scheme for efficient network utilization and QoS," IEEE/ACM Transactions, vol. 12, no. 1, pp. 161-172, Feb. 2004.

[9] D. V. Prokhorov and D. C. Wunsch II, "Adaptive Critic Designs", *IEEE Transactions on Neural Networks*, Vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[10] R. Sun,, "Individual action and collective function: From sociology to multi-agent learning", *journal of Cognitive Systems Research 2,* 2001, pp.1-3

[11] K. S. Hwang, S. W. Tan and M. C. Tsai, "Reinforcement Learning to Adaptive Control of Nonlinear System," *IEEE Transactions on Systems, Man and Cybernetics -Part B: Cybernetics,* vol. 33, no. 3, pp. 514-521, Jun. 2003.