Wireless Netw (2017) 23:1535–1548
DOI 10.1007/s11276-016-1242-y

TCP-Gvegas with prediction and adaptation in multi-hop ad hoc networks

Hong Jiang[1] · Ying Luo[1] · QiuYun Zhang[1] · MingYong Yin[2] · Chun Wu[1]

Published online: 7 March 2016

Abstract TCP Vegas performance can be improved since its rate-based congestion control mechanism could proactively avoid possible congestion and packet losses in multi-hop ad hoc networks. Nevertheless, Vegas cannot make full advantage of available bandwidth to transmit packets since incorrect bandwidth estimates may occur due to frequent topology changes caused by node mobility. This paper proposes an improved TCP Vegas based on the grey prediction theory, named TCP-Gvegas, for multi-hop ad hoc networks, which has the capability of prediction and self-adaption, as well as three enhanced aspects in the phase of congestion avoidance. The lower layers' parameters are considered in the throughput model to improve the accuracy of theoretical throughput. The prediction of future throughput based on grey prediction is used to promote the online control. The optimal exploration method based on *Q-Learning* and Round Trip Time quantizer are applied to search for the more reasonable changing size of congestion window. Besides, the convergence analysis of grey prediction by using the Lyapunov's second method proves that a shorter input data length of prediction implies a faster convergence rate. The simulation results show that the TCP-Gvegas achieves a substantially higher throughput and lower delay than Vegas in multi-hop ad hoc networks.

Keywords TCP-Gvegas · Cross throughput model · Grey prediction

✉ Hong Jiang
jianghong@swust.edu.cn

[1] School of Information, South West University of Science and Technology, Mianyang, China

[2] Institute of Computer Application, China Academy of Engineering Physics, Mianyang, China

1 Introduction

Transmission control protocol (TCP) in multi-hop ad hoc networks has recently attracted substantial research attentions [1–3]. TCP-Vegas is a typical enhanced TCP that uses a rate-based congestion control mechanism [4]. Generally speaking, TCP-Vegas has good performance in ad hoc networks for three reasons. First, the transmission rate is adjusted carefully by comparing with the estimated rate. Secondly, Vegas halves the congestion window (cwnd) size by identifying whether the retransmission packets belong to the current stage of congestion control. Thirdly, it emphasizes packet delay rather than packet loss as the criterion to determine the transmission rate. However, it only calculates the expected throughput using the round-trip time (RTT), which may not characterize the real throughput of the whole network due to the overlook of lower layer parameters. Besides, Vegas changes its current cwnd value based on the previous network situation, which means that this mechanism incurs latency to the computation and thus leads to the loss of accuracies and adaptations in rapidly changing environments.

Due to the potential improvements in performance, multiple studies have been made to cope with the problems of Vegas in wireless networks. An improved mechanism Vegas-W was proposed in [5], in which the cwnd is extended by a fraction with a rate control timer in the sending process, and the probing schemes have been changed to increase the window upon receiving more than one acknowledgment (ACK). However, the interaction analysis between TCP and lower protocols is needed for further improvement. Cheng etc. presented a threshold control mechanism with cross-layer response approach in [6] for improving TCP Vegas performance in IEEE 802.11 wireless networks. A model [7] based bandwidth estimation algorithm for 802.11 TCP data transmissions was developed by using feedback information from receivers.

Although these methods attained some improvements, all of them still have the inherent weaknesses of conventional TCP that stems from deferred calculations. Specifically, the calculations only reflect the past network conditions rather than the present and future ones, which makes them unsuitable for rapid changing environments. In this paper, we are inspired from the prediction algorithm [8], where the available duration of links is predicted to construct an efficient topology.

In this paper, we propose an improved TCP version based on TCP Vegas and Grey prediction named Gvegas, to deal with the problem of real throughput prediction and online congestion control. In addition, unlike other typical TCP variants, our scheme only needs revisions at the sender without involving intermediate nodes. To our best knowledge, this is the first work to consider both prediction and decision in congestion avoidance (CA) stage by predicting network condition before collision. Specifically, our contributions are summarized as follows:

In order to make the basis of congestion control more precisely for online congestion control in multihop ad hoc networks, we improve the theoretical and actual throughput model with cross-layer information sharing and grey prediction, respectively.

We model the congestion control decision as a MDP to choose the congestion window changing factor more effectively and adaptively. In addition, the action is quantified according to network state with round trip time.

We provide the performance analytic models of the improved congestion control over TCP Vegas, and utilize simulations and theoretical analysis to demonstrate the improved scheme can significantly promote the TCP performance in terms of both throughput and delay.

The remainder of this paper is organized as follows. We briefly describe some related works in Sect. 2. The main idea of the improved mechanism is introduced in Sect. 3. Section 4 formulates the cross layer model of expected throughput. Section 5 introduces the actual throughput prediction based on grey theory. Section 6 describes the RTT quantification and the exploration of cwnd changes based on Q-learning, in addition, the detailed algorithm and corresponding complexity analysis of space and time are also discussed. Section 7 numerically examines the Gvegas's performance along with the Vegas and Newreno. Finally, Sect. 8 provides the conclusions of this paper.

2 Related works

TCP is the most widely applied transport layer protocol for reliable data transmission. Although there are many TCP versions with various congestion control schemes for different environments, such as TCP Newreno, TCP Vegas etc., some researchers have demonstrated that TCP Vegas can achieve higher throughput, fewer retransmissions and less delay than Newreno [9], because Vegas implements a more precise congestion avoidance mechanism based on packet delay rather than the packet loss. However, Vegas fails to make full use of available bandwidth to transmit packets since incorrect bandwidth estimates may occur due to unstable conditions in multihop wireless networks.

To detect congestion more reliably, several TCP throughput models have been proposed [10–12]. Samios et al. [10] developed a model to estimate the throughput of a Vegas flow as a function of packet loss rate, RTT, and protocol parameters. Similarly, a steady-state throughput of TCP Newreno was transferred as a function of RTT and loss behavior in [11]. The simulation results in [10, 11] show that TCP throughput model combining both packet loss and delay can achieve higher throughput. Furthermore, the throughput model of Vegas in [10] was extended by [12], which combined with the TCP segment information and TCP acknowledgement (ACK) to maximize the TCP throughput. Although these researches achieve the TCP performance improvements, they don't fully concern the performance improvement from congestion control in wireless ad hoc environments.

Congestion control is another most significant way to improve TCP performance in wireless ad hoc networks [13]. An intelligent congestion control technique was proposed in [14] by using a neural network (NN) to regulate reliable data traffic in ad hoc wireless mesh networks. Badarla et al. [15] mainly concentrated on adapting to the changing network conditions and appropriately updating the congestion window by learning algorithm without relying on any network feedbacks. The above congestion control methods obtain some enhancements to some degree, however, they seldom explicitly consider the cross-layer network information to improve TCP performance.

Cross-layer designs allow information sharing among different layers in order to improve the functionality [13], in which some lower parameters could be considered when performing the cross layer design. Cheng et al. [6] proposed a congestion control scheme with dynamic threshold and cross layer response, in which the network performance was improved by the adaptation of congestion window size in accordance with the estimated buffer length. In [16], some typical cross layer factors were analyzed and modeled, and the TCP optimization was formulated as a Markov Decision Process (MDP). Furthermore, Xie et al. [17] presented a TCP pacing protocol by cross-layer measurement and analysis on the lower bound of the contention window to reduce packet burstiness in wireless networks.

In addition to the above schemes, TCP performance could be improved by status prediction through probabilistic approaches. For example, prediction strategies such as increasing the hold time of the topology for static scenario or increasing the HELLO generation rate for mobile scenario were proposed to promote the TCP efficiency [18]. Different from [18], congestion window was adjusted based on the estimation of network state by using the feedback information from the receiver [19].

Although, there are various cross-layer improved researches to improve the performance of multi-hop wireless networks, most of them always try to solve the basis of congestion and its control decision separately and make congestion control decision corresponding to past congestion control status which cannot really reflect the environmental conditions on time. In order to achieve overall and online optimization, this paper will promote the TCP performance towards the direction of prediction and joint cross-layer optimization.

3 Gvegas formulation

The Vegas mechanism uses the difference presented by (1) [4] between the expected ($v\_expect\_$) and actual ($v\_actual\_$) throughput to decide how to change cwnd:

$$\begin{aligned} diff &= v\_expect\_ - v\_actual\_ \\ &= \left(\frac{WindowSize}{BaseRTT} - \frac{SentData}{ActualRTT}\right) \cdot BaseRTT \end{aligned} \tag{1}$$

Specifically, the cwnd is either changed by adding or subtracting one packet or remains unchanged according to *diff* as follows:

$$cwnd = \begin{cases} cwnd + 1, & diff < v\_\alpha \\ cwnd - 1, & diff > v\_\beta \\ unchanged, & other \end{cases} \tag{2}$$

where v_α and v_β are corresponding threshold values [4].

Gvegas has two parts as shown in Fig. 1. The first one, which contains expected throughput, grey prediction and residual modification model, makes the congestion control more accurate. The other one, which comprises quantification and reinforcement learning model, makes the cwnd adaptive to network changes. In the first part, $v\_expect\_$ and $v\_actual\_$ in (1) are replaced with expected and predicted values, respectively. The expected model uses cross-layer parameters to obtain throughput, and the prediction uses grey theory to predict the future throughput. In the second part, we quantify RTT into cwnd changing size, and a Q-learning algorithm is used to explore the optimal changing size for different cwnd phases.

3.1 Overview of grey prediction

Grey theory is a system that focuses on the uncertainty problems of small samples and poor information [20] which consists of two parts: grey prediction and grey decision. A grey system represents the cognitive degree for environment data and is widely used due to its simple and information imperfect characteristics. Typically, there are five categories of grey prediction including: (1) time series forecasting, (2) calamity forecasting, (3) seasonal calamity forecasting, (4) topological forecasting and (5) systematic forecasting [21].

GM(1,1) model [22, 23], which is one of the most widely used grey prediction models, can weaken the stochastic fluctuation in original series by accumulated generating operation (AGO), so as to dig out its inherent law, and build a one order different equation model to describe the law. The model gets the undecided coefficients of the one order grey differential equation by using least squares estimation and obtains the final prediction value by using inverse accumulated generating operation.

3.2 Overview of Q-learning

Reinforcement learning (RL) [24] is a learning system aiming to maximize a long-term performance measure with typical scenario shown in Fig. 2. In RL model, one agent can interact with its environment autonomously: it observes environment *state* in $\boldsymbol{S}$, chooses one available *action* from $\boldsymbol{A}$, and receives one *reward* in $\boldsymbol{R}$ from environment. The state-action pair in $S \to A$ is a policy π^* which aims to maximize the total reward $V^{\pi}(s_t) = \sum_{t=0}^{\infty} \gamma^t r_t$, where the r_t is a received reward.

Q-learning is a widely used RL algorithm [25]. In real applications, due to existence of uncertainty, agent cannot forecast the reward and next state when an action is performed. This will result in that the agent cannot obtain perfect π^* by solving $V^{\pi}(s_t)$. Therefore, Q-learning uses $Q^{\pi}(s,a)$ to get the best policy: $V^{\pi^*}(s) = \max_{a \in A} Q^{\pi^*}(s,a)$ and the renewal of Q value is as follows:

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha\left[r_t + \gamma \max_{a'} Q_t(s',a') - Q_t(s,a)\right] \tag{3}$$

where α is the learning rate denoted as $\alpha = \frac{1}{1+visit(s,a)}$, *visit*(*s*,*a*) is the times of one station-action pair has been visited. s' and a' are the next state and action which belong to the set of $\boldsymbol{S}$ and $\boldsymbol{A}$, respectively. Q-learning typically selects the “greedy” action a which has the highest $Q(s,a)$ value with a probability ε (approximating 1) and selects a random action with probability $1-\varepsilon$.

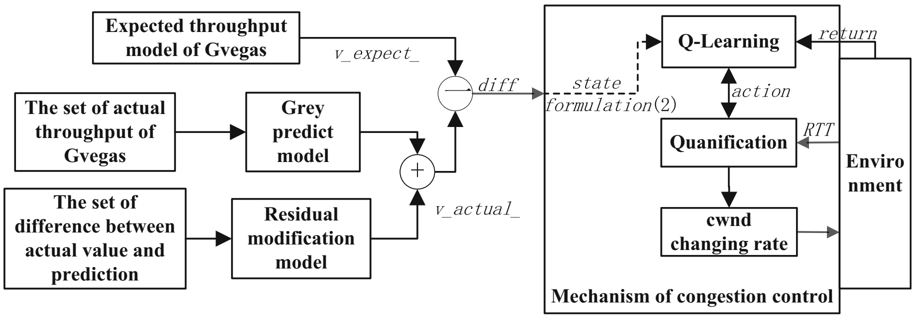

Expected throughput model of Gvegas
v_expect_
The set of actual throughput of Gvegas
Grey predict model
The set of difference between actual value and prediction
Residual modification model
v_actual_
diff
state formulation(2)
Q-Learning
action
Quanification
cwnd changing rate
RTT
return
Environment
Mechanism of congestion control

Fig. 1 The structure diagram of Gvegas

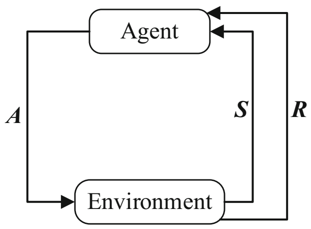


Fig. 2 The reinforcement learning scenario

4 Cross-layer model of TCP expected throughput

For simplicity of analysis, we assume that the main consideration of the throughput model involves the Transport Layer (TL) and Media Access Control (MAC) sub-layer.

4.1 TCP Vegas throughput model

A typical stochastic model for the steady-state throughput of TCP Vegas has been modeled by [10], as shown in (4), where Th_{Vegas} is the throughput received, n is the expected number of consecutive Loss Free Periods (LFP) that occurs between one slow start (SS) and another slow start period, p denotes the loss event rate, W_0 is the average of cwnd during stable backlog state, T_{TO} is the average duration of first timeout in a series of timeouts, and RTT is the average RTT.

$$Th_{Vegas} = \frac{(n+1)\left(\frac{1-p}{p} + W_0\right) + \frac{2p(2-p)}{(1-p)^2}}{N_{SS2TO}RTT + D_{TO}} \tag{4}$$

where N_{SS2TO} and D_{TO} are represented by Eqs. (5) and (6), respectively

$$N_{SS2TO} = 2\log W_0 + (n+1)\frac{1-p}{pW_0} + \left(1 + \frac{nW_0}{32}\right) + \frac{9n}{8} - \frac{11}{4} + \frac{4 - 2^{\log W_0}}{W_0} \tag{5}$$

$$D_{TO} = \left((1-p)^2 \sum_{k=1}^{6}\left(\left(2^k - 64k + 320\right)\left(2p - p^2\right)\right)^{k-1} + \frac{64}{(1-p)^2} - 321\right)T_{TO} \tag{6}$$

In order to predict the maximum achievable expected throughput for TCP-based data transmissions over IEEE 802.11 WLANs in multi-hop ad hoc networks. We extend TCP throughput model in (4) to include the characteristics of IEEE 802.11 Distributed Coordination Function (DCF) and queue management model in the subsequent subsections.

4.2 IEEE 802.11 DCF model

IEEE 802.11 DCF can be executed in either ad hoc or infrastructure-based networks and, indeed, is the typical access method implemented in most commercial wireless cards [26]. DCF contains three-times handshake as shown in Fig. 3. A sender sends a Request to Send (RTS) to a receiver after a DIFS time. Upon receiving the RTS, the receiver waits for one SIFS delay to send Clear to Send (CTS) to the sender. Then, the sender and receiver establish a link, during which the sender detects whether there is a collision before transmitting. If the answer is yes, a random back-off time will be selected by sender to build next transmission request. When the retransmission times reach the retry attempt limit, the packet will be dropped. The process time, successful delivery probability and loss probability of a packet from node to node are denoted by pt_{DCF}, dp_{DCF} and lp_{DCF}, respectively.

$$pt_{DCF} = DIFS + 3 \cdot SIFS + t_{RTS} + t_{CTS} + t_{macACK} \tag{7}$$

The value of dp_{DCF} is given by [7]

$$dp_{DCF} = (1 - D\_\tau)^{ncdf-1} \cdot (1 - BER)^{dl+dh} \tag{8}$$

where $ncdf$ indicates the number of competition channel data flow. D_τ denotes the probability of a request to send

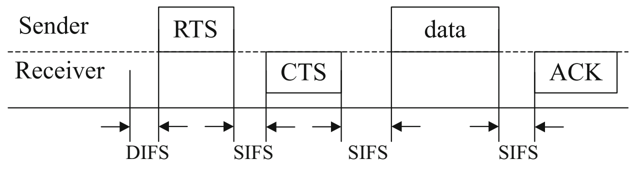


Fig. 3 The access mechanism of DCF with RTS/CTS

data. BER, dl, dh and CW_I are the bit error rate, packet size, length of packet header and initial size of cwnd, respectively, i.e.

$$D\_\tau = \frac{2}{CW_I + 1} \tag{9}$$

When the retransmission times reach the retry attempt limit rl, a packet is dropped with probability given by

$$lp_{DCF} = (1 - dp_{DCF})^{rl} \tag{10}$$

4.3 Queue management model

In this paper, we assume that Random Early Discard (RED) is used as the mechanism of queue management. The loss probability and the processing time of successfully transmitted packet at the RED stage are denoted by lp_{queue} and pt_{queue}, respectively. pt_{queue} is available in real time via the feedbacks of RED protocol. lp_{queue} is given by

$$lp_{queue} = \begin{cases} 0, & queue_t^{avg} \le queue_{t-1}^{\min} \\ 1, & queue_t^{avg} \ge queue_{t-1}^{\max} \\ \dfrac{queue_t^{avg} - queue_{t-1}^{\min}}{queue_{t-1}^{\max} - queue_{t-1}^{\min}}, & otherwise \end{cases} \tag{11}$$

where $queue_{t-1}^{\min}$ and $queue_{t-1}^{\max}$ are the minimal and maximal size of queue at time $t-1$. $queue_t^{avg}$ indicates the average queue length at time t.

4.4 The expected throughput model

We formulate the expected throughput model by joint consideration of the term Th_{Vegas} in (4) and the limitation of the maximum cwnd in a multihop ad hoc network. The throughput model is given by:

$$NV\_t = \min\left(\frac{WindowSize}{BaseRTT}, \frac{(n+1)\left(\frac{1-p^*}{p^*} + W_0\right) + \frac{2p^*(2-p^*)}{(1-p^*)^2}}{N_{SS2TO}RTT^* + D_{TO}}\right) \tag{12}$$

where N_{SS2TO} and D_{TO} are defined by (5) and (6), respectively. Besides, p^* and RTT^* are defined by

$$p^* = \left(1 - \prod_{i=1}^{hs}\left(1 - lp_{DCF}^i\right)\right) + \left(1 - \prod_{i=1}^{hs}\left(1 - lp_{queuequeue}^i\right)\right) + lp_{TCPdrop} \tag{13}$$

and

$$RTT^* = \sum_{i=1}^{hs}\left(\left(1 - lp_{DCF}^i\right)\cdot pt_{DCF}^i\right) + \sum_{i=1}^{hs}\left(\left(1 - lp_{queue}^i\right)\cdot pt_{queue}^i\right) + \left(1 - lp_{TCPdrop}\right) \cdot pt_{TCP\_ACK} \tag{14}$$

where hs represents the sum of hops of a network connection, the p in N_{SS2TO} and D_{TO} is replaced by p^*. The $lp_{TCPdrop}$ and pt_{TCP_ACK} are the drop probability and processing time of TCP ACK which could be detected in real time via feedbacks of TCP protocol. The value of pt_{DCF}, lp_{DCF} and lp_{queue} can be calculated by (7, 10) and (11).

5 Prediction of actual throughput based on grey model

Chapter 4 discusses the expected throughput in cross-layer model, however, the existing TCP variants also use the actual throughput of last stage in the Eq. (1) as the basis to control the congestion at next stage. This mechanism cannot cope with the real-time control and unpredictable dynamics of ad hoc networks. Thus, we propose a future actual throughput prediction scheme based on the GM(1,1) grey model [22, 23].

5.1 The grey prediction of throughput

We focus on predicting the actual throughput of the next stage by grey prediction GM(1,1) model [23] shown in Fig. 4. For simplicity, we assume that grey model predicts at discrete time interval called "stage". The mode predicts the future network throughput at stage $t+1$ to determine the cwnd changing size. In our model, the actual network throughputs of the last t stages are regarded as the input column $ag^o = \{ag^o(1), ag^o(2),\ldots,ag^o(t)\}$, where t represents the length of input data and its minimum length is 4

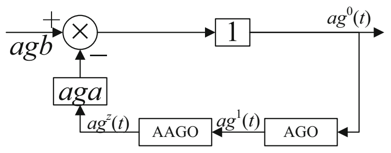


Fig. 4 Definition model of GM(1,1)

for the purpose of prediction precision [23]. Accumulated Generating Operation (AGO), which will output the sequence of ag^1, aims to eliminate the randomness of ag^o. In addition, $ag^z(t)$, created by Adjacent Average Generation Operation (AAGO), is the intermediate sequence to calculate the parameters of *aga* and *agb* as follows:

$$agb - aga \cdot ag^z(t) = ag^0(t) \tag{15}$$

where *aga* and *agb* are evolution parameter and grey action, respectively:

$$aga = \frac{\sum_{k=2}^{t} ag^z(k) \cdot \sum_{k=2}^{t} ag^0(k) - (t-1) \cdot \sum_{k=2}^{t} ag^z(k) \cdot ag^0(k)}{(t-1) \cdot \sum_{k=2}^{t} ag^z(k)^2 - \left(\sum_{k=2}^{t} ag^z(k)\right)^2} \tag{16}$$

$$agb = \frac{\sum_{k=2}^{t} ag^0(k) \cdot \sum_{k=2}^{t} ag^1(k)^2 - \sum_{k=2}^{t} ag^1(k) \cdot \sum_{k=2}^{t} ag^1(k) \cdot ag^0(k)}{(t-1) \cdot \sum_{k=2}^{t} ag^z(k)^2 - \left(\sum_{k=2}^{t} ag^z(k)\right)^2} \tag{17}$$

The throughput prediction for next stage is denoted by

$$a\hat{g}^0(t+1) = \left(ag^0(1) - \frac{agb}{aga}\right) \cdot e^{-aga \cdot t} \cdot (1 - e^{aga}) \tag{18}$$

5.2 Residual modification model

Residual modification model is used to correct the prediction error, which is given by

$$ar^0(k) = \frac{a\hat{g}^0(k) - v\_actual\_(k)}{v\_actual\_(k)}, \quad k = \{1, 2, \ldots, t\} \tag{19}$$

The original sequence of prediction error is set to be the input of grey prediction model. Therefore, the prediction of error, which is defined as $a\hat{r}^0(t+1)$, will reduce the error between the predicted and actual throughputs. Finally, the throughput prediction with residual modification is given by

$$gv\_actual\_ = a\hat{g}^0(t+1) + a\hat{r}^0(t+1) \cdot v\_actual\_(t) \tag{20}$$

5.3 Stability analysis of grey prediction

5.3.1 The condition of convergence

We analyze the stability of the grey prediction system according to Lyapunov's second method. The definition of differential function of grey prediction is given by

$$\frac{dx^{(1)}}{dt} = x^{(1)\prime} = agb - aga \cdot x^{(1)} \tag{21}$$

We define the positive definite function of Lyapunov as

$$V(x^{(1)}) = (agb - aga \cdot x^{(1)})^2 + c, (c > 0) \tag{22}$$

The time derivative of V is given by

$$V(x^{(1)})' = -2aga \cdot (agb - aga \cdot x^{(1)}) \cdot x^{(1)'} \tag{23}$$

Equation (23) can be rewritten as follows by combining with (21).

$$V(x^{(1)})' = -2aga \cdot (agb - aga \cdot x^{(1)})^2 \tag{24}$$

According to Lyapunov's second method [27], if *aga* is larger than zero, then $V(x^{(1)})'$ is negative semi-definite, which means that the Eq. (21) has asymptotic stability about zero solutions. At the same time, the value of *aga* should belong to the range $[-2/(t+1), 2/(t+1)]$ for the sake of ensuring prediction accuracy in grey model [22]. Thus, we let the value of *aga* lie in the range $(0, 2/(t+1)]$.

5.3.2 Convergence rate of grey prediction

The solution of (21) is given by [22]

$$\hat{x}^{(1)}(t+1) = \left(x^{(1)}(0) - \frac{agb}{aga}\right) \cdot \exp(-aga \cdot t) + \frac{agb}{aga} \tag{25}$$

with prediction sequence $\hat{x}^{(1)}(t+1)$. The next stage prediction of $x^{(0)}(t)$ can be calculated by

$$\hat{x}^{(0)}(t+1) = \left(x^{(1)}(0) - \frac{agb}{aga}\right) \cdot \exp(-aga \cdot t) \cdot (1 - \exp(aga)) \tag{26}$$

under the relationship of $x^{(0)}(t+1) = x^{(1)}(t+1) - x^{(1)}(t)$.

According to the definition, (26) can be rewritten as $\hat{x}^{(0)}(t+1) = \exp(aga) \cdot \hat{x}^{(0)}(t)$, which can be replaced with $x_{t+1} = g(x_t) = \exp(-aga) \cdot x_t$. We assume that the initial value of system is perturbed by ζ_0. Then, according to the Lyapunov exponential spectrum [27], the distance between original and interrupted values is by the following equation after N iterations

$$\zeta_t \cong \zeta_0 \cdot \exp(\lambda \cdot t_s) \tag{27}$$

where t_s represents continuous-time variable and λ is defined by

$$\lambda = \lim_{N \to \infty} \frac{1}{N} \sum_{i=0}^{N-1} \ln\left|\frac{df(x_i, u)}{dx}\right| \tag{28}$$

where N is the times of iterations, x_i is the prediction value of the *ith* iteration.

Finally, the evaluation of the Lyapunov exponential spectrum of (28) is given by

$$\begin{aligned}\lambda &= \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1}\ln\left|\frac{df(x_i,u)}{dx}\right| \\ &= \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1}\ln\left|\frac{dg(x_i,u)}{dx}\right| \\ &= \lim_{N\to\infty}\frac{1}{N}\sum_{i=0}^{N-1}\ln|\exp(-aga)| \\ &= -aga\end{aligned} \tag{29}$$

From (29), (27) can be rewritten as $\zeta_t \cong \zeta_0 \cdot \exp(-aga \cdot t_s)$. When $aga \in (0,2/(t+1)]$, the Lyapunov exponential spectrum $\lambda \in [-2/(t+1),0) < 0$. It means that the phase volume of system is contracted in this direction, and the movement is stable in this direction. Specifically, the distance between original and interrupted values decreases exponentially as $|\lambda|$ increases. In other words, the convergence rate is exponential in the length of input data.

6 Mechanism of congestion control

In the process of transmission, we try to make a full use of available network capacity and follow the conditions and principles of Vegas simultaneously by combining quantification with Q-learning. From formulation (1, 12) and (20), the final diff is calculated by

$$\begin{aligned}diff &= v\_expect\_ - v\_actual\_ \\ &= (NV\_t - gv\_actual\_)\cdot BaseRTT\end{aligned} \tag{30}$$

6.1 RTT quantization

In order to take more proper actions according to the circumstances, we combine the throughput and the RTT quantization to control the cwnd change sizes, instead of only using the criterion of throughput. At the same time, the RTT is quantified to different degrees as the size of cwnd changes, which is described by

$$cwnd=\begin{cases} cwnd + bl+1, & RTT \le \min RTT \\ cwnd + bl + 1 - \dfrac{1}{(\max RTT - \min RTT)^2}\cdot(RTT - \min RTT)^2, & \min RTT < RTT \le \max RTT \\ wnd + bl, & \max RTT < RTT \end{cases} \tag{31}$$

In order to overcome the conservation of cwnd changes in traditional Vegas mechanism, *minRTT* and *maxRTT* in Eq. (31) are the minimal and maximal RTTs at each congestion control stage, respectively.

The random parameter *bl* is uniformly chosen in the interval $[m_{sf}\cdot\gamma, n_{sf}\cdot\gamma]$, where γ value is -1, 0 or 1 according to throughput state,m_{sf} and n_{sf} are span factor which could determine the width of the range. Besides, we formulate the phase of CA as a MDP to explore the optimal value of *bl* for cwnd changing phases for the purpose of improving the accuracy of congestion control.

6.2 Q-learning exploration model

In the Q-learning model, we let the state S_t be the different stages of cwnd changes. The action associated with a state is the value selection for *bl*. The return of $R_i(s,a_i)$ is set to be the value of the actual throughput. The purpose of the Q-learning is to choose the optimal action that maximizes the following Q function [28].

$$\begin{aligned}Q_{t+1}(s_t,a_t) = [1-\alpha_t]\cdot Q_{t+1}(s_t,a_t) + \alpha_t \\ \cdot\left[R_{t+1} + (q\gamma)\cdot \max_a Q_t(s_{t+1},a)\right]\end{aligned} \tag{32}$$

where $q\gamma \in (0,1]$ is a discount factor that denotes the relative importance of present and future rewards. Q converges to Q^* with probability 1 [28] when the learning rate α_n meets the condition $\sum_{i=1}^{\infty}\alpha_{t(i,s,a)} = \infty, \sum_{i=1}^{\infty}\left(\alpha_{t(i,s,a)}\right)^2 < \infty$, where $t(i,s,a)$ means that the action a is selected in state s in the ith iteration. The optimal strategy of s_t is defined by $\pi^*(s_t) = \underset{s_t}{\operatorname{argmax}}\, Q^*(s_t,a)$.

For a tradeoff between exploration and exploitation, we utilize the $\varepsilon - greedy$ [29] action selection strategy, in which the optimal action is selected with probability of ε.

6.3 Algorithm and complexity analysis

The algorithm of improved congestion avoidance control with prediction and self-adaption are summarized in Algorithm 1.

Algorithm 1: congestion avoidance control algorithm with prediction and self-adaption

```
Initialize $ag^0[t]=ag^1[t]=ag^z[t]=Q(s_t,a_t,r_t)=0$
**while** cwnd > threshold in each RTT **do**
Evaluate the expected theoretical throughput based on (12);
Update the actual throughput for $ag^0[t]$ by using network feedback;
Update the error of throughput for $ar^0[t]$ according to (19);
**for** $i = 1 \rightarrow t$ **do**
    Update the sequence of $ag^1$, $ag^z$, $ar^1$, $ar^z$;
**end for**
Evaluate parameter *aga* and grey action *agb* according to (16) and (17), respectively;
Get the parameter *ara* and grey action *arb* in the same way;
    Calculate the future throughput $\hat{ag}^0(t+1)$ based on (18) and obtain the prediction error
$\hat{ar}^0(t+1)$ in the same way;
Evaluate the throughput prediction with revision *gv_actual_* based on (20) and update the latest
*diff* by function (30) to determine the latest state $s_t$ based on equation (2);
Get the latest cwnd changing size according to (31);
Choose an updating policy $a_t = \arg\max_{s_{t+1}} Q^*(s_{t+1}, A)$ with probability $\varepsilon$ or a random policy with
probability $1-\varepsilon$;
**end while**
```

From Algorithm 1, the space and time complexities of Gvegas mainly rely on the ones of the grey model and Q-learning model. The space and time complexities of the grey prediction can be denoted by $O_{sg}(t)$ and $O_{tg}(t-1)$, respectively. Similarly, the space and time complexity of Q-learning can be denoted by $O_{qg}(|S| \cdot |A|)$ and $O_{qg}(|A|)$ [29]. Therefore, the overall space complexity of Gvegas is $O(|S| \cdot |A|)$, and the time complexity is $O(\max(|A|, (t-1)))$. In addition, there is no deployment difficulty for Gvegas which is realized only on the sender without any changes at routers and receivers.

7 Simulation results and discussions

We implement Gvegas in NS2 and verify its performance along with TCP Vegas and Newreno under different length of input data ($t = 4, 5, 6, 7, 8$) in three simulation scenarios: "chain", "Reference Point Group Mobility Model (RPGM)" and "Wireless Mobile Ad Hoc Network (WMAHN)" scenarios, and each performance metric of scenario is computed by averaging results of 500 simulation runs. We assume that the Internet layer, MAC layer and queue manager use IP, IEEE 802.11 and RED protocols, respectively. All scenarios have the same MAC layer

Table 1 MAC sub-layer parameters

Parameters	Value
Minimize of cwnd size	31
Maximize of cwnd size	1023
Slot time	20 μs
SIFS	10 μs
Data length	144 bit
Header length	48 bit
Short retry limit	7
Long retry limit	4

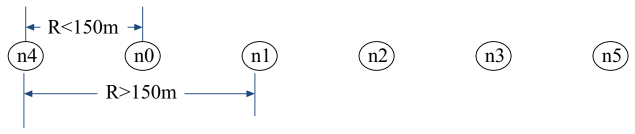


Fig. 5 Positions information of nodes

parameters given in Table 1. We set the span factor $m_{sf} = 1$, and $n_{sf} = 3$. The learning factor $q\gamma$ and learning rate α_t are equal to 0.95 and 0.3 [30], respectively. In addition,ε is set to be 0.8. One TCP flow in Gvegas, Vegas or Newreno is always transmitted from sender nodes 4 to receiver node 5 throughout the simulation in each scenario,

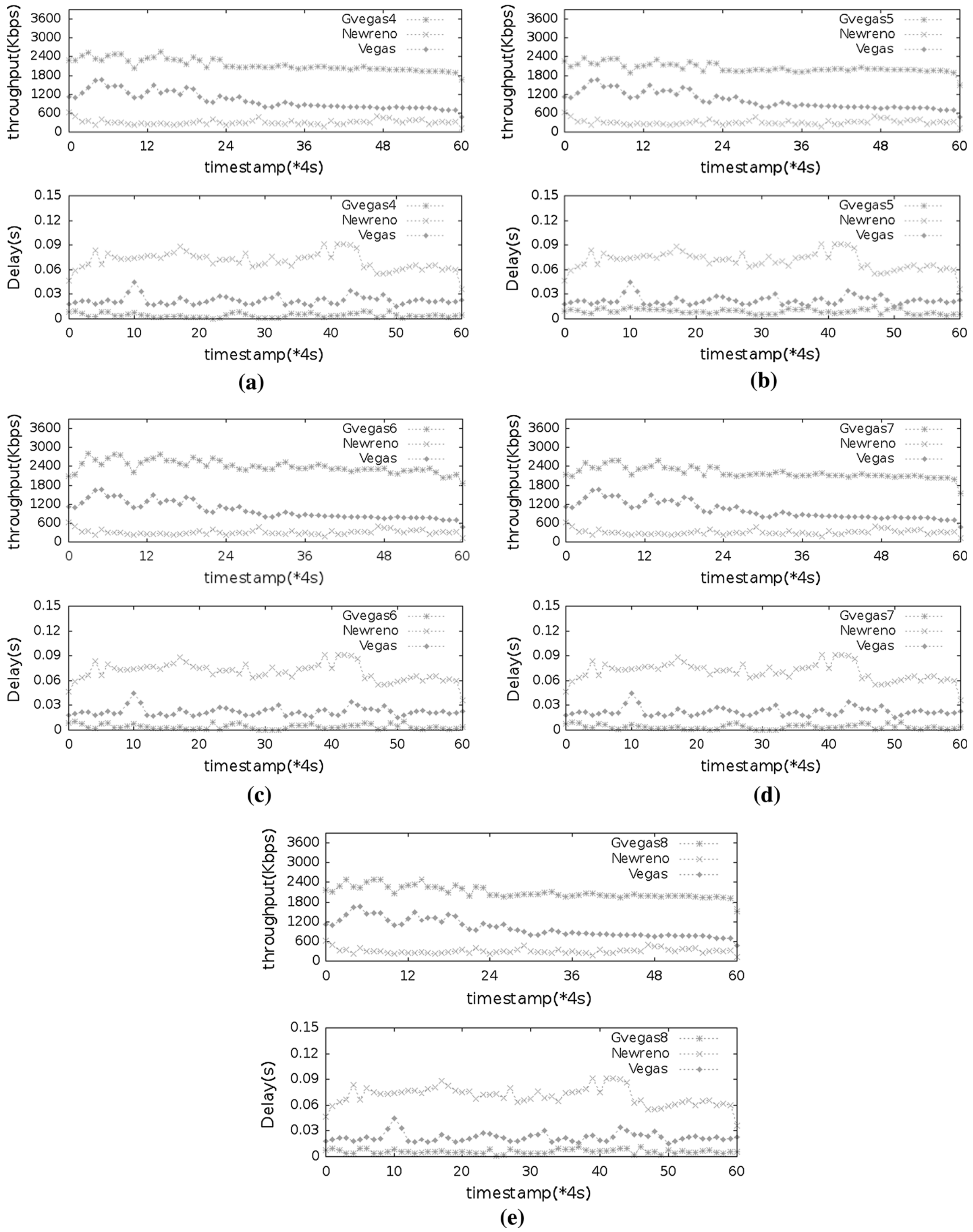


Fig. 6 The average simulation results of multi-hop chain network. **a** $t = 4$, **b** $t = 5$, **c** $t = 6$, **d** $t = 7$, **e** $t = 8$

and the end-to-end throughput and delay between node 4 and node 5 are collected to be the performance analysis parameters by gathering the simulation information from those two nodes. Each chain scenario simulation has a duration of 240 and 600 s for each RPGM and WMAHN scenario. For the purpose of better display, each four seconds is denoted as a timestamp in multi-hop chain scenario, and ten seconds for both RPGM and WMAHN scenario.

7.1 Multi-hop chain wireless network scenario

We assume that there are six nodes that have the identical transmission range of 150 meters and initial positions of those nodes are indicated in Fig. 5. In this simulation, a linear multi-hop chain is kept by all nodes with random velocities in a range [5, 8] m/s. The average throughput and delay of 500 runs among Gvegas, Vegas and Newreno are displayed by Fig. 6. In addition, the expectations of throughput and delay of 500 runs each with 240 s simulation of different prediction length of $t = \{4, 5, 6, 7, 8\}$ are shown by Fig. 7.

From Fig. 6, we compare the performance in terms of throughput and delay of Gvegas with those of Vegas and Newreno for different $t = \{4, 5, 6, 7, 8\}$. We can see that both the average throughput and delay of Gvegas are significantly improved compared with Vegas and Newreno, irrespective of the values of t in Fig. 6. From Fig. 7, we can further see that the throughput of Gvegas is improved more than doubled and fivefold in Vegas and Newreno, respectively. In addition, the delay of Gvegas is shorter by 42 and 90 % on average than Vegas and Newreno, respectively.

7.2 RPGM scenario

RPGM is a typical rescue type of ad hoc mobile group mode, in which each group has a logic center, and other nodes are equally distributed around the center. The group movement is represented by change of the center, including the position, velocity, direction and accelerated speed, etc.

Let a group have six nodes, each having the same transmission range of 80 m in a 800 × 600 m area. We assume that the group is formed by these six nodes randomly distributed in a sub-area of 200 × 200 m. The group's velocity is uniformly randomly chosen in [1, 3] m/s with a random pause time when reaching a destination. The average throughput and delay of 500 simulation runs among Gvegas, Vegas and Newreno in RPGM are displayed by Fig. 8. In addition, the expectations of throughput and delay of 500 runs each with 600 s simulation of different prediction length of $t = \{4, 5, 6, 7, 8\}$ are shown by Fig. 9.

From Fig. 8, we can see that the performance of Gvegas is greatly better than that of Vegas and Newreno in terms of both average throughput and transmission delay in RPGM. In more detail, Fig. 9 shows that Gvegas attains more throughputs than Vegas and Newreno by about 22 and 52 % on average for each $t = \{4, 6, 5, 7, 8\}$, and shorter delay by about 60 and 82 %, respectively. Because Gvegas has the capability to foresee the environment conditions in the near future, Gvegas can take reasonable actions before collision or congestion based on the prediction. As demonstrated in Sect. 5.3, the shorter the input data length of prediction is, the faster the convergence rate is. The simulation results in RPGM scenario further confirm that the curves of both throughput and delay in forecast sequence of $t = 4$ are more smoothly than other prediction length.

7.3 WMAHN scenario

WMAHN is a self-configuring, dynamic network in which nodes are free to move. We assume that ten nodes are randomly distributed in a square area of 300 × 300 m area and each node has the same transmission range of 80 m. In addition, we let the node's velocity be uniformly randomly chosen in [1, 5] m/s with a random pause time in [0,5]s when arriving at a destination. The average throughput and delay of 500 simulation runs among Gvegas, Vegas and Newreno in WMAHN are displayed by Fig. 10. Furthermore, the expectations of throughput and delay of 500 runs each

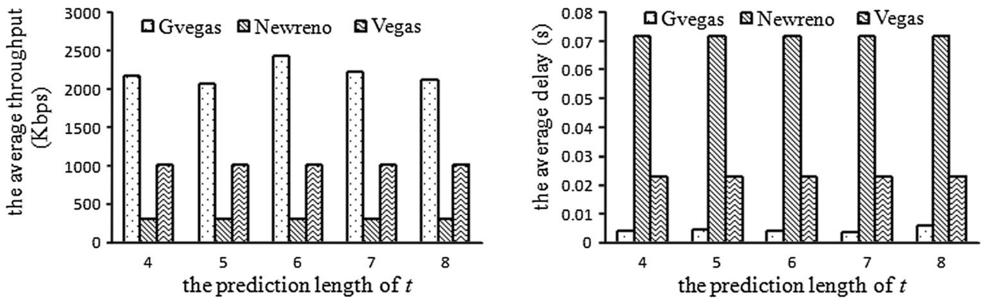


Fig. 7 The expectation of throughput and delay of chain of different prediction length

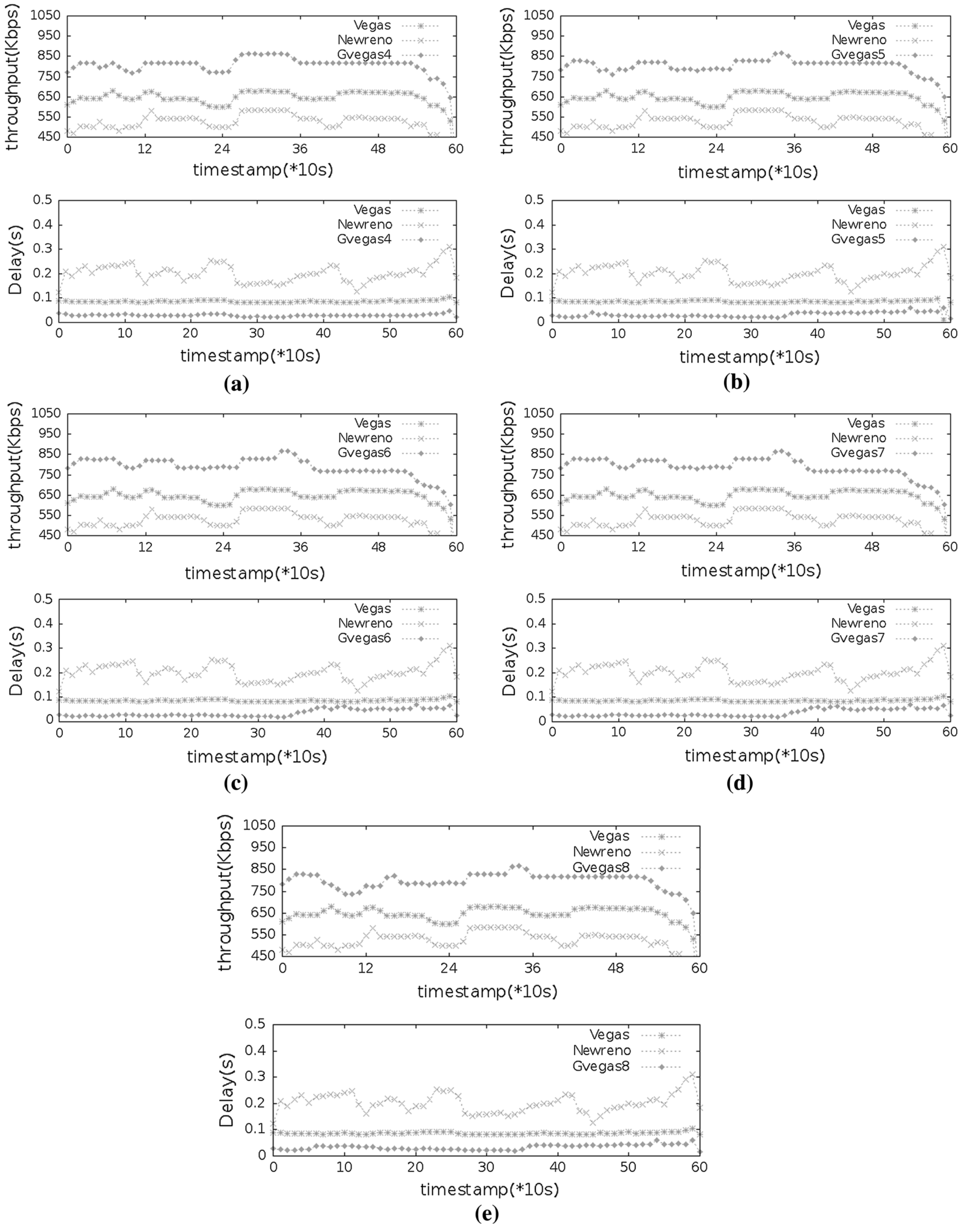


Fig. 8 The average simulation results of group ad hoc network. **a** $t = 4$, **b** $t = 5$, **c** $t = 6$, **d** $t = 7$, **e** $t = 8$

with 600 s simulation of different prediction length of $t = \{4, 5, 6, 7, 8\}$ are shown by Fig. 11.

From Fig. 10, we can see that the performance of Gvegas is obviously better than that of Vegas and Newreno after a few simulation times. At the beginning of simulation, Gvegas is not stable since the perception of dynamic environment is not enough for Q-learning and Grey predictor. But, the throughput and delay of Gvegas become better than Vegas and Newreno through exploration about 100 s. This is further explained in Fig. 11, which shows that the expectation of throughput of Gvegas is better than that of Vegas and Newreno by about 60 % and five times

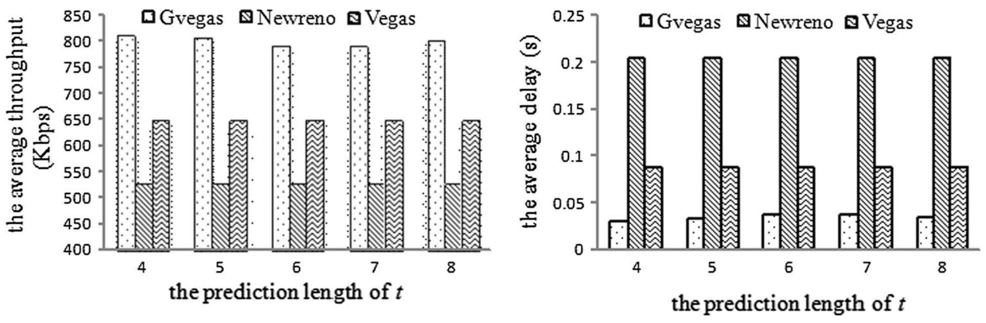
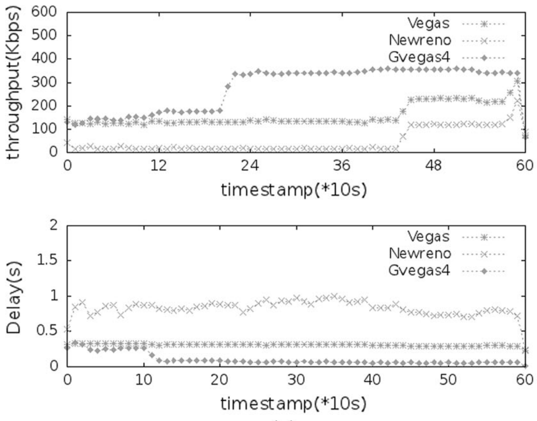
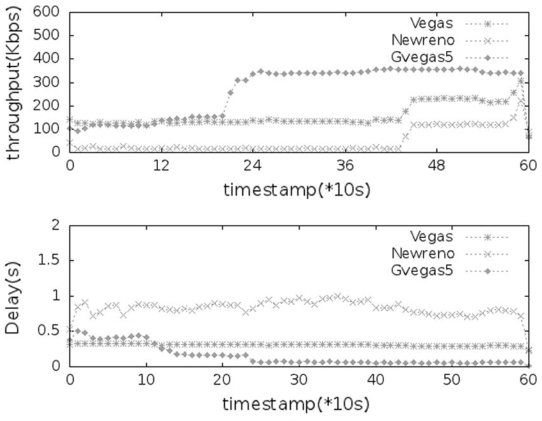


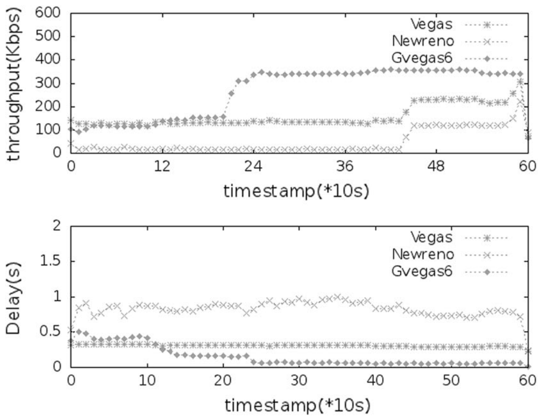
Fig. 9 The expectation of throughput and delay of RPGM of different prediction length



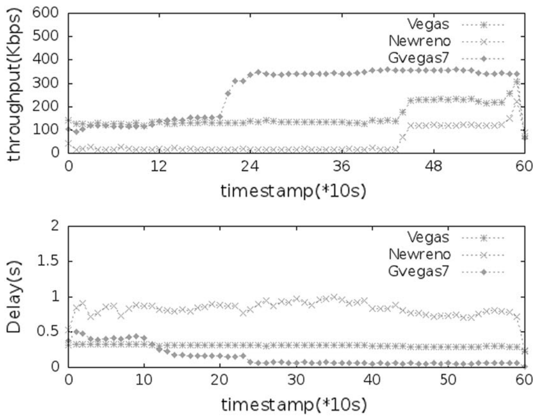
(a)



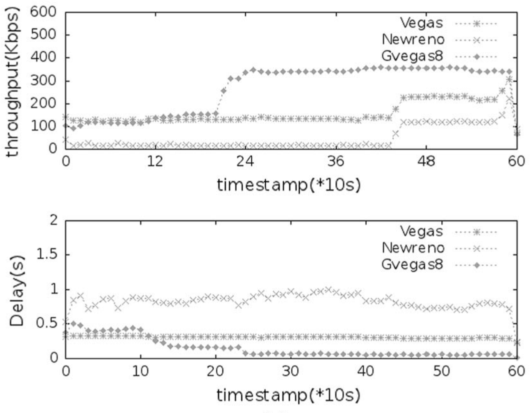
(b)



(c)



(d)



(e)

Fig. 10 The average simulation results of mobile ad hoc network. **a** $t = 4$, **b** $t = 5$, **c** $t = 6$, **d** $t = 7$, **e** $t = 8$

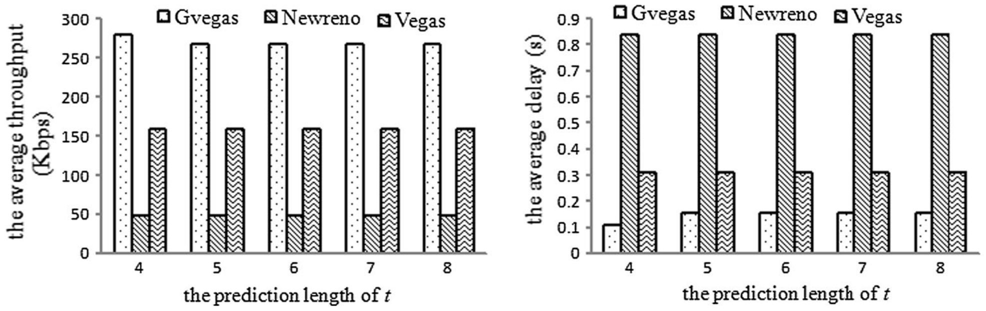


Fig. 11 The expectation of throughput and delay in WMAHM of different prediction length

for each $t = \{4, 6, 5, 7, 8\}$, and delay shorter by about 60 and 80 %, respectively. Besides, the simulation results confirm that the curves of both throughput and delay in forecast sequence of $t = 4$ are more smoothly than other prediction length.

8 Conclusions

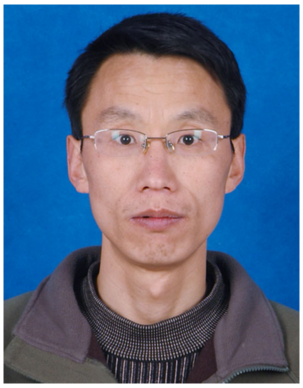
This paper uses the cross-layer throughput model to calculate the achievable throughput and predict the throughput of the next time stage based on grey theory for real-time congestion control. In addition, the quantization of RTT is performed by the optimal action exploration of Q-learning. The convergence condition and rate of grey model based on Lyapunov's second method have shown the effectiveness of the improved method. Simulation results show that the performance of our proposed scheme greatly outperforms the Vegas and Newreno in terms of both average throughput and delay in multi-hop ad hoc networks. From simulation results and convergence analysis, we suggest that the prediction length $t = 4$ is a reasonable choice for typical scenarios.

Acknowledgments This work was supported in part by National Nature Science Foundation of China under Grant 61379005, State Administration of Science, Technology and Industry for National Defence under Grant B3120133002, China Academy of Engineering Physics(CAEP) project under Grant 2015A0403002, and the Robot Technology Used for Special Environment Key Laboratory of Sichuan Province under Grant 13zxtk07. The authors would like to thank the editor and anonymous reviewers for their thorough review and valuable comments and suggestions to improve the quality of the paper.

References

1. Cai, Y., Jiang, S., Guan, Q., et al. (2013). Decoupling congestion control from TCP (semi-TCP) for multi-hop wireless networks. *EURASIP Journal on Wireless Communications and Networking, 1*, 1–14.
2. Al-Jubari, A. M., Othman, M., Ali, B. M., et al. (2013). An adaptive delayed acknowledgment strategy to improve TCP performance in multi-hop wireless networks. *Wireless Personal Communications, 69*(1), 307–333.
3. Majeed, A., Abu-Ghazaleh, N. B., Razak, S., et al. (2012). Analysis of TCP performance on multi-hop wireless networks: A cross layer approach. *Ad Hoc Networks, 10*(3), 586–603.
4. Brakmo, L. S., O'Malley, S. W., & Peterson, L. L. (1994). TCP Vegas: New techniques for congestion detection and avoidance. In *Proceedings of the conference on communications architectures, protocols and applications (ACM SIGCOMM)* (pp. 24–35).
5. Ding, L., Wang, X., Xn, Y., et al. (2008). Improve throughput of TCP-Vegas in multi-hop ad hoc networks. *Computer Communications, 31*(10), 2581–2588.
6. Cheng, R. S., Deng, D. J., Chao, H. (2011). Congestion control with dynamic threshold adaptation and cross-layer response for TCP over IEEE 802.11 wireless networks. In *Proceedings of the international conference on wireless information networks and systems (WINSYS)* (pp. 95–100).
7. Yuan, Z., Venkataraman, H. M., Muntean, G. M. (2012). A novel bandwidth estimation algorithm for IEEE 802.11 TCP data transmissions. 2012 Wireless Vehicular Communications and Networks (WCNC) (pp. 377–382).
8. Guan, Q., Yu, F. R., Jiang, S., et al. (2010). Prediction-based topology control and routing in cognitive radio mobile ad hoc networks. *IEEE Transactions on Vehicular Technology, 59*(9), 4443–4452.
9. Ghaffari, A. (2015). Congestion control mechanisms in wireless sensor networks: A survey. *Journal of Network and Computer Applications, 52*, 101–115.
10. Samios, C. B., & Vernon, M. K. (2003). Modeling the throughput of TCP Vegas. *ACM SIGMETRICS Performance Evaluation Review, 31*(1), 71–81.
11. Parvez, N., Mahanti, A., & Williamson, C. (2010). An analytic throughput model of TCP Newreno. *IEEE/ACM Transactions on Networking, 18*(2), 448–461.
12. Xie, H., Boukerche, A., De Grande, R., et al. (2015). Towards a distributed TCP improvement through individual contention control in wireless networks. *IEEE International Conference on Communications (ICC), 2015*, 5602–5607.
13. Fu, B., Xiao, Y., Deng, H., et al. (2014). A survey of cross-layer designs in wireless networks. *IEEE Communications Surveys and Tutorials, 16*(1), 110–126.
14. Al Islam, A. B. M. A., & Raghunathan, V. (2015). ITCP: An intelligent TCP with neural network based end-to-end congestion control for ad-hoc multi-hop wireless mesh networks. *Wireless Networks, 21*(2), 581–610.
15. Badarla, V., & Murthy, C. S. R. (2011). Learning-TCP: A stochastic approach for efficient update in TCP congestion window in ad hoc wireless networks. *Journal of Parallel and Distributed Computing, 71*(6), 863–878.
16. Xie, H., Pazzi, R. W., & Boukerche, A. (2012). A novel cross layer TCP optimization protocol over wireless networks by Markov Decision Process. *IEEE Global Communications Conference (GLOBECOM), 2012*, 5723–5728.

17. Xie, H., & Boukerche, A. (2015). TCP-CC: Cross-layer TCP pacing protocol by contention control on wireless networks. *Wireless Networks, 21*(4), 1061–1078.
18. Mezzavilla, M., Quer, G., & Zorzi, M. (2014). On the effects of cognitive mobility prediction in wireless multi-hop ad hoc networks. *IEEE International Conference on Communications (ICC), 2014*, 1638–1644.
19. Wang, J., Dong, P., Chen, J., et al. (2013). Adaptive explicit congestion control based on bandwidth estimation for high bandwidth-delay product networks. *Computer Communications, 36*(10), 1235–1244.
20. Liu, S., Forrest, J., & Yang, Y. (2012). A brief introduction to grey systems theory. *Grey Systems: Theory and Application, 2*(2), 89–104.
21. Xie, N., & Liu, S. (2009). Discrete grey forecasting model and its optimization. *Applied Mathematical Modelling, 33*(2), 1173–1186.
22. Kayacan, E., Ulutas, B., & Kaynak, O. (2010). Grey system theory-based models in time series prediction. *Expert Systems with Applications, 37*(2), 1784–1789.
23. Chen, C. I., & Huang, S. J. (2013). The necessary and sufficient condition for GM(1,1) grey prediction model. *Applied Mathematics and Computation, 219*(11), 6152–6162.
24. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, 4*, 237–285.
25. Marco, M., Lorenza, G., Michele, R., et al. (2015). Distributed Q-learning for energy harvesting heterogeneous networks. *IEEE International Conference on Communications (ICC), 2015*, 2006–2011.
26. Alonso-Zárate, J., Crespo, C., Skianis, C., et al. (2012). Distributed point coordination function for IEEE 802.11 wireless ad hoc networks. *Ad Hoc Networks, 10*(3), 536–551.
27. Paunonen, L., & Zwart, H. (2013). A Lyapunov approach to strong stability of semigroups. *System and Control Letters, 62*(8), 673–678.
28. Yau, K. L. A., Komisarczuk, P., & Teal, P. D. (2012). Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues. *Journal of Network and Computer Applications, 35*(1), 253–267.
29. Shiang, H. P., & Van der Schaar, M. (2010). Online learning in autonomic multi-hop wireless networks for transmitting mission-critical applications. *IEEE Journal on Selected Areas in Communications, 28*(5), 728–741.
30. Xu, X., Liu, C., & Hu, D. (2011). Continuous-action reinforcement learning with fast policy search and adaptive basis function selection. *Soft Computing, 15*(6), 1055–1070.



Hong Jiang received the Ph.D degree in School of Communication and Information Engineering from University of Electronic Science and Technology of China. He is a full professor at the South West University of Science and Technology of China. His current interests include the cross layer Qos support in adhoc networks, and intelligent learning in cognitive radio networks.

Ying Luo received the Master degree in School of Information, South West University of Science and Technology of China. Her current research interests include cross layer optimization in wireless networks, and cooperative communication.

QiuYun Zhang received the Master degree in School of Information, South West University of Science and Technology of China. His current research interests include MAC layer optimization in wireless networks.

MingYong Yin received the M.S. degree in School of Computer Science from Fudan University. His current interests include resource management in communication networks.

Chun Wu received the Ph.D. degree in School of Telecommunications Engineering from XiDian University, China. He is an associate professor at the South West University of Science and Technology of China. His current interests include resource management in adhoc networks, and intelligent learning in cognitive radio networks.