On Modeling Datagram Congestion Control Protocol and Random Early Detection using Fluid-Flow Approximation

HIROYUKI HISAMATSU HIROYUKI OHSAKI MASAYUKI MURATA
Graduate School of Information Science and Technology
Osaka University
JAPAN
hisamatu@ist.osaka-u.ac.jp oosaki@ist.osaka-u.ac.jp murata@ist.osaka-u.ac.jp

Abstract — In this paper, we model DCCP congestion control mechanism and RED as independent discrete-time systems using fluid-flow approximation. By interconnecting DCCP connections and RED routers, we model the entire network as a feedback system called *DCCP/RED*. We then analyze the steady state performance and the transient state performance of DCCP/RED. Specifically, we derive the packet transmission rate of DCCP connections, the packet transmission rate, the packet loss probability, and the average queue length of the RED router in steady state. Moreover, we investigate the parameter region where DCCP/RED operates stably by linearizing DCCP/RED around its equilibrium point. We also evaluate the transient state performance of DCCP/RED in terms of ramp-up time, overshoot, and settling time. Consequently, we show that the stability and the transient state performance of DCCP/RED degrade when the weight of the exponential weighted moving average, which is one of RED control parameters, is small. To solve this problem, by adding changes to the function with which RED determines the packet loss probability, we propose RED-IQI (RED with Immediate Queue Information), as an applications of our analytic result. We analyze the transient state performance of the feedback system DCCP/RED-IQI where DCCP connections and RED-IQI routers are interconnected. Consequently, we show that DCCP/RED-IQI has significantly better transient state performance than DCCP/RED.

Key-Words: DCCP (Datagram Congestion Control Protocol), RED (Random Early Detection), Control Theory, Fluid-flow Approximation, Steady State Performance, Transient State Performance

1. Introduction

In recent years, real-time applications, such as video streaming, IP telephone, TV conference, and network game, become popular rapidly by increasing speed of the network, or the rising demand for multimedia applications [1]. Generally, either UDP (User Datagram Protocol) [2] or TCP (Transmission Control Protocol) [3] has been used as a transport layer protocol for real-time applications. Since the Internet is a best-effort network where multiple users share the network bandwidth, all network applications need to have a mechanism for adapting to congestion status of the network. However, UDP is simply a protocol for datagram transfer, and does not have a mechanism for controlling network congestion. Hence, when a real-time application uses UDP as a transport layer protocol, it is necessary for the application to implement a certain congestion control mechanism at an application layer for preventing congestion collapse of the network [4].

On the contrary, TCP has a mechanism for adjusting the packet transmission rate according to the available bandwidth of the network by performing congestion control between source and destination hosts. However, TCP is a transport layer protocol originally designed for data transfer applications that can tolerate a certain amount of transmission delay [5]. Since the congestion control mechanism of TCP is the AIMD (Additive Increase and Multiplicative Decrease) window flow control, the packet transmission rate from a source host fluctuates at the time scale of approximately round-trip time. Although such fluctuation is not a problem when using TCP with non-realtime applications such as data transfer applications, it becomes a serious problem in real-time applications such as video streaming [5].

DCCP (Datagram Congestion Control Protocol) is therefore proposed as a new transport layer protocol for real-time applications [6]. DCCP performs congestion control between source and destination hosts, and an application using DCCP can choose the type of congestion control mechanisms. Currently, “TCP-like congestion control profile” [7] that performs congestion control similar to TCP, and “TFRC congestion control profile” [8] that performs congestion control similar to TFRC (TCP Friendly Rate Control) are proposed.

In the TCP-like congestion control profile, an AIMD window control is performed as with TCP [7]. The AIMD window control additively increases the window size (i.e., the number of packets that can be transmitted in a round-trip time) until a source host detects network congestion. If congestion in the network is detected, a source host multiplicatively decreases the window size. Therefore, the packet transmission rate of DCCP using the TCP-like congestion control profile fluctuates at the time scale of approximately round-trip time. Hence, for instance, DCCP with the TCP-like congestion control profile is suitable for a streaming application that buffers a large amount of data at a destination host [7].

On the contrary, in the TFRC congestion control profile, variation of the packet transmission rate caused by the TCP-like congestion control profile is prevented, and congestion control is performed so that the network bandwidth is fairly shared with other competing TCP connections [8]. In DCCP with the TFRC congestion control profile, a destination host primarily performs congestion control. Namely, in the TFRC congestion control profile, the destination host detects network congestion and notifies it of a source host. The source host adjusts the packet transmission rate from a source host based on the congestion information (e.g., *packet loss event rate*) notified from the destination host. For instance, DCCP with the TFRC congestion control profile is suitable for a streaming application that buffers a small amount of data at a destination host [8].

Whereas DCCP performs congestion control between source and destination hosts, AQM (Active Queue Management) mechanisms that perform congestion control at routers in the network have been capturing the spotlight in recent years [4, 9]. A representative AQM mechanism is RED (Random Early Detection) [10], which probabilistically discards an arriving packet. With RED, as compared with the conventional DropTail, the average queue length (i.e., the average number of packets in the buffer) of the router can be kept small, and high throughput can be achieved [10, 11]. In particular, keeping the average queue length small is effective in decreasing the end-to-end transmission delay. Hence, it is expected that an AQM mechanism is effective for real-time applications.

In the literature, there exist not many theoretical studies on DCCP [12]. On the contrary, in the literature, many studies on the congestion control mechanism of TCP, which is adopted in the TCP-like congestion control profile of DCCP, have extensively performed [13-20]. In particular, characteristics of the mixed environment of TCP connections and RED routers have been extensively studied. For instance, in [14-16], the congestion control mechanism of TCP and RED are modeled as independent discrete-time systems. The entire network is then modeled as a feedback system where TCP connections and the RED router are interconnected. By applying control theory, the steady state performance and the transient state performance of the TCP congestion control mechanism and RED are analyzed. Moreover, in [13, 17, 18], the TCP congestion control mechanism and RED are modeled as independent continuous-time systems, and the steady state performance of RED is analyzed. In [20], it is shown that the transient state performance and the robustness of RED improve, when the function with which RED determines the packet loss probability is changed to a concave function to the average queue length.

Although characteristics of the mixed environment of TCP congestion control mechanism and RED have been sufficiently investigated, characteristics of the mixed environment of TFRC congestion control mechanism and RED have not been sufficiently studied [5, 21-23]. In particular, to the best of our knowledge, transient state performance of TFRC has not been fully investigated. In [21], fairness between TCP-friendly rate control mechanism and TCP in steady state is evaluated with simulations and traffic measurements of the Internet. Moreover, in [5], fairness between TFRC and TCP is evaluated by simulation. The transient state performance of a TCP-friendly rate control mechanism is also evaluated. However, these studies assume that all routers are DropTail and the effect of the interaction between TFRC connections and RED routers has not been fully investigated [22, 23].

In this paper, we therefore model DCCP congestion control mechanism and RED as independent discrete-time systems by using the modeling approach in [14-16]. We then analyze the steady state performance and the transient state performance of DCCP/RED. Specifically, we derive the packet transmission rate of DCCP connections, the packet transmission rate, the packet loss probability, and the average queue length of the RED router in steady state. Moreover, we investigate the parameter region where DCCP/RED operates stably by linearizing DCCP/RED around its equilibrium point. We also evaluate the transient state performance of DCCP/RED in terms of ramp-up time, overshoot, and settling time. Consequently, we shown that the stability and the transient state performance of DCCP/RED degrade when the weight of the exponential weighted moving average, which is one of RED control parameters, is small. To solve this problem, by adding changes to the function with which RED determines the packet loss probability, we propose RED-IQI (RED with Immediate Queue Information), as an applications of our analytic result. We analyze the transient state performance of the feedback system DCCP/RED-IQI, where DCCP connections and RED-IQI routers are interconnected. Consequently, we show that DCCP/RED-IQI has significantly better transient state performance than DCCP/RED.

The organization of this paper is as follows. In Section 2, we briefly explain the overview of DCCP and its two congestion control profiles: the TCP-like congestion control profile and the TFRC congestion control profile. In Section 3, we model DCCP congestion control mechanism and RED as independent discrete-time systems. By interconnecting these models, we obtain DCCP/RED model, the model of the entire network. In Section 4, we derive the packet transmission rate of DCCP connections, the packet transmission rate, the packet loss probability, and the average queue length of the RED router in steady state. In Section 5, we analyze the transient state performance of DCCP/RED by linearizing DCCP/RED around its equilibrium point. Moreover, in Section 6, we present several numerical examples and show quantitatively how the steady state performance and the transient state performance of DCCP/RED change with the bottleneck link bandwidth and the propagation delay of the network. We also show that the stability and the transient state performance of DCCP/RED degrade when the weight of the exponential weighted moving average is small. In Section 7, we propose RED-IQI by adding changes to the function with which RED determines the packet loss probability. We then analyze the transient state performance of DCCP/RED-IQI. Finally, in Section 8, we conclude this paper and discuss future works.

2. DCCP (Datagram Congestion Control Protocol)

DCCP is a transport layer protocol designed for realtime applications [6]. The reliable data transfer is not guaranteed in DCCP. Namely, even if a packet is discarded in the network, a source host does not retransmit a lost packet.

In DCCP, applications using DCCP can choose a congestion control mechanism by specifying the congestion control profile. The identifier called CCID (Congestion Control IDentifier) is assigned to each congestion control profile supported by DCCP. At the time of connection establishment, source and destination hosts of DCCP exchange information on supported CCIDs, and negotiate the congestion control profile used during the data transfer. Moreover, DCCP supports ECN [24] and ECN Nonce [25], which are mechanisms by which the router explicitly notifies the congestion occurrence of the source host. Currently, CCID2 (TCP-like congestion control profile) and CCID3 (TFRC congestion control profile) are supported as congestion control profiles [7, 8].

In the TCP-like congestion control profile, the AIMD window control is performed similarly to TCP [7]. In the AIMD window control, a source host additively increases the window size (i.e., the number of packets that can be transmitted in a round-trip time) until the source host detects network congestion. If the network congestion is detected, the source host multiplicatively decreases the window size. However, the TCP-like congestion control profile of DCCP differs from TCP congestion control in the following four points.

First, the congestion control of DCCP is performed also to ACK packets from a destination host to a source host using the *ACK Ratio mechanism* [7]. The transmission rate of ACK packets that a destination host returns to a source host is determined by the ACK Ratio. Specifically, when the ACK Ratio is R, the destination host of DCCP will send one ACK packet back to the source host per R data packets received from a source host.

Second, since DCCP is an unreliable transport layer protocol, a source host of DCCP does not retransmit a packet [7]. In the congestion control mechanism of TCP, when a packet is discarded, the source host identifies whether it is a retransmission packet. However, such procedure is not performed in DCCP.

Third, a destination host of DCCP can notify the cause of a packet loss of a source host [7]. This is realized by the *Data Dropped option* contained in ACK packets from a destination host to a source host. For instance, the destination host can notify it of the source host whether the packet loss is resulted from bit error of the transmission link or buffer overflow at the destination host.

Fourth, the TCP-like congestion control profile of

DCCP does not perform the flow control; i.e., only the AIMD window control is performed. The buffer management of a destination host, which is performed by TCP congestion control mechanism using the advertising window, is not performed in DCCP.

On the contrary, in the TFRC congestion control profile, TCP-friendly congestion control that can fairly share bandwidth with competing TCP congestion control is performed, avoiding variation of the packet transmission rate [8]. In DCCP with the TFRC congestion control profile, congestion control is primarily performed at a destination host. Namely, in DCCP with the TFRC congestion control profile, a destination host detects network congestion, and it is notified of a source host. The source host adjusts the packet transmission rate from a source host based on the congestion information (e.g., packet loss event rate) notified from the destination host. The TFRC congestion control profile of DCCP differs from TFRC congestion control in the following point.

In the TFRC congestion control profile, a destination host can notify the cause of a packet loss of a source host [8]. This is realized similarly to the TCP-like congestion control profile using the Data Dropped option contained in ACK packets.

3. Modeling DCCP and RED

In this paper, we model DCCP congestion control mechanism and RED as independent discrete-time systems with a time slot of Δ. We model the entire network as a single feedback system where DCCP connections and RED routers are interconnected. First, we model the congestion control mechanism of DCCP as a discrete-time system, where the input is the packet arrival rate at a destination host and the output is the packet transmission rate from a source host. Next, we model RED as a discrete-time system, where the input is the packet arrival rate and the output is the packet transmission rate.

Figure 1 shows the analytic model used in this paper. N DCCP connections share the single bottleneck link. All DCCP connections' two-way propagation delays are equal, which are denoted by τ. The bottleneck link bandwidth is denoted by μ. We denote four control parameters of RED by max_p (maximum packet loss probability), max_{th} (maximum threshold), min_{th} (minimum threshold), and w_q (weight of exponential weighted moving average). Furthermore, RED buffer size is denoted by L. Table 1 shows the definition of symbols used in this analysis.

In this analysis, we introduce a concept of *the packet arrival rate at a destination host* notified of a source host by ACK packets, to unify the input and the output of the models to the packet arrival/transmission rate. Since information on the arrival status of packets at a destination host is included in ACK packets, a source host can estimate the packet arrival rate at a destination host.

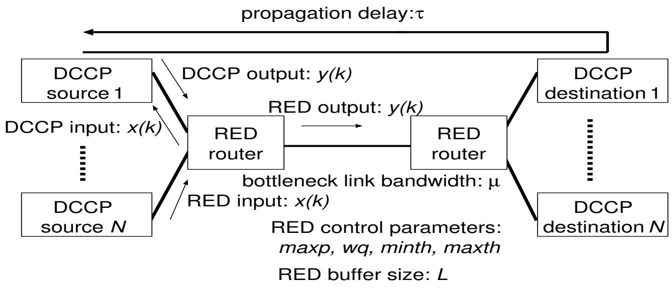


Fig. 1: Analytic model

TABLE 1
DEFINITION OF SYMBOLS

network parameters	
N	number of DCCP connections
τ	two-way propagation delay of DCCP connection
μ	bottleneck link bandwidth
L	buffer size of RED router
Δ	time slot
DCCP parameters	
$w(k)$	window size of CCID2
t_{RTO}	retransmission timer of CCID2
$p_e(k)$	packet loss event rate of CCID3
$R(k)$	round-trip time
RED parameters	
max_p	maximum packet loss probability
min_{th}	minimum threshold
max_{th}	maximum threshold
w_q	weight of exponential weighted moving average
$q(k)$	current queue length
$\overline{q}(k)$	average queue length
$p(k)$	packet loss probability

In this analysis, we assume the followings; since DCCP is mainly used for real-time applications, it is assumed that a source host always has data to transfer. When the packet loss probability of the network is small and DCCP congestion control works appropriately, DCCP operates in the congestion avoidance phase. Therefore, DCCP with the TCP-like congestion control profile is assumed to operate in the congestion avoidance phase.

First, we model change of the DCCP window size. The packet loss probability in the network is denoted by p, and the DCCP window size is denoted by w. Change of the DCCP window size is given by [26]

$$\begin{aligned} w \quad \leftarrow \quad & w + (1-p)\frac{1}{w} - p\,(1 - p_{TO}(w,p))\,\frac{1}{2}\frac{4\,w}{3} \\ & - p\,p_{TO}(w,p)\left(\frac{4\,w(k)}{3} - 1\right), \end{aligned}$$

where $p_{TO}(w,p)$ is the probability that DCCP detects the packet loss by the timeout mechanism when the window size is w and the packet loss probability is p [27]:

$$\begin{aligned} & p_{TO}(w,p) = \\ & \qquad \frac{(1-(1-p)^3)\,(1+(1-p)^3\,(1-(1-p)^{w-3}))}{(1-(1-p)^w)}. \end{aligned}$$

$p(k)$ is defined as the packet loss probability at slot k in the network, $R(k)$ the DCCP round-trip time, and $w(k)$ the DCCP window size. The packet loss probability of the network that a source host detects at slot k is given by $p(k - \frac{R(k)}{\Delta})$. Suppose that ACK packets are not discarded due to congestion on the path from a destination host to a source host, the ACK Ratio value converges to 1 [7]. Hence, the DCCP window size $w(k+1)$ at slot $k+1$ is approximately given by

$$\begin{aligned} w(k+1) \simeq w(k) + & \frac{w(\delta)}{R(k)}\,\Delta\Big\{(1-p(\delta))\frac{1}{w(k)} \\ & -p(\delta)(1 - p_{TO}(w(\delta), p(\delta)))\frac{2\,w(k)}{3} \\ & -p(\delta)\,p_{TO}(w(\delta), p(\delta))(\frac{4\,w(k)}{3} - 1)\Big\}, \end{aligned} \tag{1}$$

where $\delta \equiv k - R(k)/\Delta$.

The packet arrival rate at a destination host $x(k)$ is determined by the past packet transmission rate of a source host and the past packet loss probability in the network, $y(\delta)$ and $p(\delta)$.

$$x(k) \quad = \quad (1 - p(\delta))y(\delta) \tag{2}$$

Thus, the DCCP packet transmission rate is given by the following equation from change of the DCCP window size given by Eq. (1).

$$\begin{aligned} & y(k+1) = \\ & \quad y(k) + \Delta\frac{x(k)}{y(k)R(k)^2} - \frac{2}{3}\Delta y(k)\,\{y(\delta) - x(k)\} \\ & \quad \times\left\{1 - p_{TO}(x(\delta)\,R(\delta), 1 - \frac{x(k)}{y(\delta)})\right\} \\ & \quad -\Delta\left\{\frac{4}{3}y(k) - \frac{1}{R(k)}\right\}\{y(\delta) - x(x)\} \\ & \quad \times p_{TO}(x(\delta)\,R(\delta), 1 - \frac{x(k)}{y(\delta)}) \end{aligned} \tag{3}$$

Next, we model the congestion control mechanism of DCCP with the TFRC congestion control profile as a discrete-time system. The input $x(k)$ of DCCP with the TFRC congestion control profile is the packet arrival rate at the destination host notified of the source host at slot k. Moreover, the output $y(k)$ is the packet transmission rate from a source host at slot k.

The packet loss event rate at slot k is defined by $p_e(k)$, and the DCCP connection's round-trip time $R(k)$. Suppose that the source host receives an ACK packet at slot k. In this case, the DCCP source host changes the transmission rate $y(k+1)$ at slot $k+1$ as [28]

$$y(k+1) \quad = \quad \min\left(X(p_e(k), R(k)), 2\,x(k)\right), \tag{4}$$

where $X(p_e(k), R(k))$ is given by

$$\begin{aligned} & X(p_e(k), R(k)) = \\ & \quad \frac{1}{R(k)\sqrt{\frac{2p_e(k)}{3}} + t_{RTO}\left(3\sqrt{\frac{3p_e(k)}{8}}p_e(k)(1+32p_e(k)^2)\right)}, \end{aligned}$$

where t_{RTO} is the TCP retransmission timer, and is can be approximated by $4R(k)$ [28].

Supposing that a RED router discards a packet randomly with the probability p, the packet loss event rate p_e measured by DCCP and the packet loss probability p at a RED router satisfy the following relation:

$$\begin{aligned} \frac{1}{p(k)} \quad = \quad & 1 \times \sum_{i=1}^{M}\Big((1-p_e(k))^{i-1}\,p_e(k)\Big) \\ & + \sum_{i=M+1}^{\infty}\Big(i\,(1-p_e(k))^{i-1}\,p_e(k)\Big), \end{aligned}$$

where M is the number of packets $M(= R(k)\,y(k))$ that arrive at the RED router during a round-trip time.

Finally, we model the RED router as a discrete-time system. The input $x(k)$ is the packet arrival rate at the RED router at slot k. Moreover, the output $y(k)$ is the packet transmission rate from the RED router at slot k.

We define μ as the bottleneck link bandwidth and $p(k)$ as the probability that the RED router discards packets. Since the packet arrival rate at the RED router is $x(k)$, the packet transmission rate from the RED router is given by $(1 - p(k))\,x(k)$. Furthermore, since the maximum packet transmission rate from the RED router is limited by the output link bandwidth, the maximum of $y(k)$ is limited by the bottleneck link bandwidth μ. Hence, the output $y(k)$ of RED is given by [26]

$$y(k) = \min((1-p(k))\,x(k), \mu). \tag{5}$$

The current queue length of RED at slot k is denoted by $q(k)$, and the average queue length is denoted by $\overline{q}(k)$.

When the buffer size of the RED router is L, the current queue length $q(k+1)$ at slot $k+1$ is given by [26]

$$q(k+1) \;=\; \min\left[\max\left\{q(k) + \left(x(k) - \mu\right)\Delta, 0\right\}, L\right] \qquad (6)$$

Let q be the current queue length of RED, and $\overline{q}$ be the average queue length of RED. RED updates the average queue length $\overline{q}$ for every packet receipt as [10]

$$\overline{q} \leftarrow (1 - w_q)\,\overline{q} + w_q\, q. \qquad (7)$$

Since the packet arrival rate at slot k is $x(k)$, the average queue length $\overline{q}(k)$ at slot $k+1$ is approximately given by [26]

$$\overline{q}(k+1) \;\simeq\; \overline{q}(k) + x(k)\,\Delta\, w_q(q(k) - \overline{q}(k)). \qquad (8)$$

RED determines the packet loss probability $p_b(k)$ from its average queue length $\overline{q}(k)$ [10] as

$$p_b(k) \;=\; \begin{cases} 0 & \text{if } \;\overline{q}(k) < min_{th} \\ \dfrac{max_p}{max_{th} - min_{th}}(\overline{q}(k) - min_{th}) & \\ & \text{if } \; min_{th} \leq \overline{q}(k) < max_{th} \\ 1 & \text{if } \;\overline{q}(k) \geq max_{th}. \end{cases} \qquad (9)$$

Finally, the RED router discards arriving packets with the probability $p_a(k)$ determined by

$$p_a(k) = \frac{p_b(k)}{1 - count \times p_b(k)}, \qquad (10)$$

where $count$ is the number of packets arrived at the router since the last packet discarded. Since the packet loss probability $p(k)$ in the RED router is the average of $p_a(k)$, it is given by [10]

$$p(k) \;=\; \frac{2 p_b(k)}{1 + p_b(k)}. \qquad (11)$$

Note that using the current queue length $q(k)$ of RED, a DCCP connection's round-trip time at slot k is given by

$$R(k) \;=\; \frac{q(k)}{\mu} + \tau.$$

4. Steady State Analysis

In what follows, we analyze the steady state performance of DCCP/RED utilizing analytic models constructed in Section 3. Specifically, we derive the packet transmission rate of DCCP connections, the packet transmission rate, the packet loss probability, and the average queue length of RED in steady state. In Section 6, we will validate our approximate analysis by comparing numerical examples with simulation ones.

Since the congestion control mechanism of DCCP with the TCP-like congestion control profile is the AIMD window control, the window size oscillates when the feedback delay is not negligible. Consequently, the packet transmission rate never converges to a fixed value. Note that the output from our DCCP model with the TCP-like congestion control profile represents not an instantaneous value of the oscillating packet transmission rate, but the expected value of the packet transmission rate.

The packet transmission rate of DCCP and RED in steady state ($k \to \infty$) are denoted by y_D^* and y_R^*, respectively. Let N be the number of DCCP connections. We can numerically obtain y_D^* and y_R^* by solving equations $y(k+1) = y(k) = y_R^*$, $x(k) = \frac{y_R^*}{N}$ (Eq. (3)), $y(k+1) = y(k) = y_R^*$, and $x(k) = N\, y_D^*$ (Eq. (5)). Focusing on the input x_R^* and the output y_R^* of a RED router, we have the following relation

$$y_R^* \;=\; (1 - p^*)\, x_R^*, \qquad (12)$$

where p^* is the packet loss probability at the RED router in steady state. We can obtain p^* by solving Eq. (12) for p^*. Furthermore, from Eqs. (9) and (11), we can easily obtain the average queue length $\overline{q}^*$ of the RED router.

5. Transient State Analysis

We then analyze the transient state performance of DCCP/RED by linearizing the discrete-time model around its equilibrium point.

First, we focus on the feedback system where DCCP connections with the TCP-like congestion control profile and RED routers are interconnected. The state of DCCP and RED is determined by the packet arrival rate $x_D(k)$ at the destination host (notified by a destination host via ACK packets) at slot k, the packet transmission rates $y_D(k) \cdots y_D(k - \frac{R(k)}{\Delta})$ from the source host, the packet arrival/transmission rate of the RED router at slot k, $x_R(k)$ and $y_R(k)$. We introduce a state vector $\mathbf{x}(k)$ that are composed of differences between each state variable at slot k and its equilibrium value:

$$\mathbf{x}(k) \;\equiv\; \begin{bmatrix} x_D(k) & - & x_D^* \\ y_D(k) & - & y_D^* \\ & \vdots & \\ y_D(k - \frac{R(k)}{\Delta}) & - & y_D^* \\ x_R(k) & - & x_R^* \\ y_R(k) & - & y_R^* \end{bmatrix}$$

We focus on state transition between slot k and slot $k+1$. Although all discrete models (Eqs. (1)–(3), (5)–(11)) in our analysis are nonlinear, they can be written in the following matrix form by linearizing them around their equilibrium values x_D^*, y_D^*, x_R^*, and y_R^*.

$$\mathbf{x}(k+1) \quad = \quad \mathbf{A}\mathbf{x}(k), \qquad (13)$$

where $\mathbf{A}$ is the state transition matrix of the state vector from $\mathbf{x}(k)$ to $\mathbf{x}(k{+}1)$. The eigenvalues of the state transition matrix $\mathbf{A}$ determine the transient state performance (i.e., convergence performance to the equilibrium point) of the discrete-time systems given by Eqs. (1)–(3), (5)–(11) [29]. Let $\lambda_i(1 \leq i \leq \frac{R(k)}{\Delta} + 3)$ be the eigenvalues of the state transition matrix $\mathbf{A}$. The maximum absolute value of eigenvalues (*maximum modulus*) determines the stability and the transient state performance of the feedback system around its equilibrium point [29]. It is known that the smaller the maximum modulus is, the better the transient state performance becomes. It is also known that the system is stable if the maximum modulus is less than 1.0.

Next, we focus on the feedback system where DCCP connections with the TFRC congestion control profile and RED routers are interconnected. The state of DCCP with the TFRC congestion control profile and RED are determined by the packet arrival rate $x_D(k)$ at the destination host at slot k , the packet transmission rates $y_D(k) \cdots y_D(k - \frac{R(k)}{\Delta})$ from the source host, and the packet arrival/transmission rate of RED,at slot k, $x_R(k)$ and $y_R(k)$. Hence, the state vector $\mathbf{x}(k)$ that are composed of differences between each state variable at slot k and its equilibrium value is given by (14).

We assume that the DCCP destination host sends an ACK packet to its source host every n slots. We focus on state transition between slot k and slot $k + n$. Although all discrete models in our analysis (Eqs. (4)–(11)) are nonlinear, they can be written in the following matrix form by linearizing them around their equilibrium values x_D^*, y_D^*, x_R^*, and y_R^*.

$$\mathbf{x}(k+n) \quad = \quad \mathbf{A}\,\mathbf{B}^{n-1}\mathbf{x}(k), \qquad (14)$$

where $\mathbf{A}$ is the state transition matrix of the state vector from $\mathbf{x}(k)$ to $\mathbf{x}(k+1)$ when the DCCP source host receives an ACK packet (Eq. (4)). Moreover, $\mathbf{B}$ is the state transition matrix of the state vector from $\mathbf{x}(k)$ to $\mathbf{x}(k+1)$ when the DCCP source host does not receive any ACK packet (i.e., $x(k+1) = x(k)$). $\mathbf{A}\,\mathbf{B}^{n-1}$ is the state transition matrix of the state vector from $\mathbf{x}(k)$ to $\mathbf{x}(k+n)$. The eigenvalues of the state transition matrix determine the transient state performance (i.e., the convergence performance to the equilibrium point) of the discrete-time system given by Eqs. (4)–(11).

6. Numerical Examples

In this section, by presenting some numerical examples, we show quantitatively how the steady state performance and the transient state performance of DCCP/RED change according to the bottleneck link bandwidth and the propagation delay of the network. Furthermore, we validate our approximate analysis by comparing analytic results with simulation ones.

Unless explicitly stated, in the following numerical examples and simulations, values shown in Tab. 2 are used as control parameters and system parameters. We performed simulation using ns-2 for the network topology shown in Fig. 1. In this network, the link between two RED routers is the bottleneck, so that we focus on the packet loss probability and the average queue length of the upstream RED router. We run simulation for 150 [s] and used simulation result of the last 100 [s] for measuring DCCP connections' packet transmission rates and the packet loss probability of the RED router. We repeated simulation 10 times and measured averages of the DCCP connections' packet transmission rates and the packet loss probability of the RED router.

First, we focus on the steady state performance of DCCP/RED. We show the DCCP packet transmission rate for different settings of the bottleneck link bandwidth in Fig. 2. Here, we configure the DCCP connection's two-way propagation delay to $\tau = 50$ and $\tau = 100$ [ms]. Figure 2(a) shows results for DCCP with the TCP-like congestion control profile. Figure 2(b) shows for DCCP with the TFRC congestion control profile.

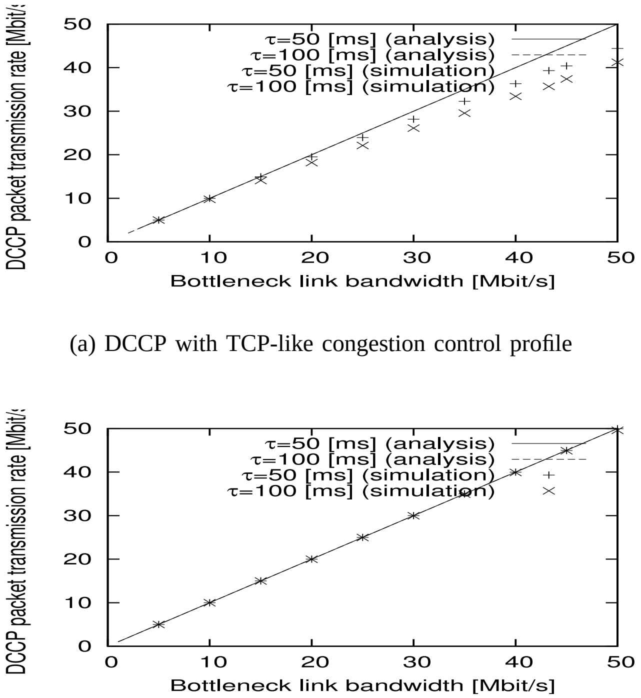
These figures indicate that the DCCP packet transmission rate increases as the bottleneck link bandwidth increases. Moreover, we compare analytic results with simulation ones. In DCCP with the TCP-like congestion control profile, some errors are observed between analytic results and simulation ones in the region where bottleneck link bandwidth is large. In other region, analytic results and simulation ones coincide closely.

We show the packet loss probability of the RED router for different settings of the bottleneck link bandwidth in Fig. 3. The DCCP connection's two-way propagation delay is configured to $\tau = 50$ and $\tau = 100$ [ms]. Figure 3(a) shows results for DCCP with the TCP-like congestion control profile. Figure 3(b) shows for DCCP with the TFRC congestion control profile. These figures show that the packet loss probability of RED decreases rapidly as the bottleneck link bandwidth increases. Moreover, it indicates that analytic results and simulation ones coincide with sufficient accuracy.

TABLE 2

PARAMETERS USED IN NUMERICAL EXAMPLE AND SIMULATION

network parameters		
number of DCCP connections	N	10
two-way propagation delay of DCCP connection	τ	50, 100 [ms]
access link bandwidth		10 μ [Mbit/s]
packet length of DCCP connection		1000 [byte]
RED parameters		
maximum packet loss probability	max_p	0.1
minimum threshold	min_{th}	20 [packet]
maximum threshold	max_{th}	100 [packet]
weight of exponential weighted moving average	w_q	0.002



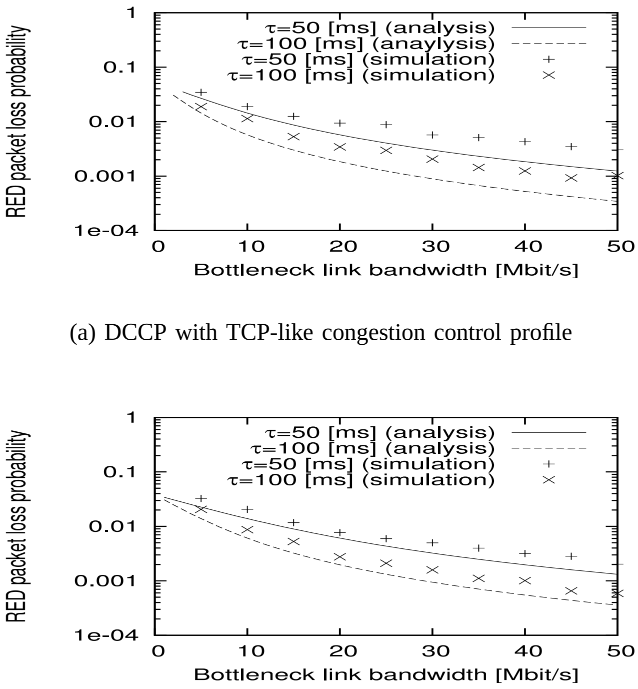
(a) DCCP with TCP-like congestion control profile

(b) DCCP with TFRC congestion control profile

Fig. 2: DCCP/RED steady state performance (DCCP packet transmission rate)

Next, we focus on the transient state performance of DCCP/RED. Figure 4 shows the maximum modulus of the state transition matrix ($\mathbf{A}$ or $\mathbf{A}\,\mathbf{B}^{n-1}$) of DCCP/RED for different settings of the bottleneck link bandwidth. Figure 4(a) shows results for DCCP with the TCP-like congestion control profile (Eq. (3)). Figure 4(b) shows results for DCCP with the TFRC congestion control profile (Eq. (4)). In these figures, the weight w_q of the exponential weighted moving average of RED is configured to 0.0002, 0.002 and 0.02. Moreover, the number of DCCP connections is $N = 1$, and the two-way propagation delay of DCCP connection is $\tau =$ 10 [ms].

These figures show that the maximum modulus increases as the bottleneck link bandwidth increases. This means that the transient state performance of DCCP/RED degrades as the bottleneck link bandwidth increases. Moreover, it can be found that the maximum modulus increases as the weight w_q of the exponential weighted moving average of RED becomes small. This can be explained as follows. The time for the average queue length of RED following change of the network state increases as the weight w_q of the exponential weighted moving average becomes small. Hence, it



(a) DCCP with TCP-like congestion control profile

(b) DCCP with TFRC congestion control profile

Fig. 3: DCCP/RED steady state performance (RED packet loss probability)

becomes slow that the packet loss probability of RED follows change of the network state. Namely, setting w_q to be a small value has the same effect with increasing the feedback delay of the entire network.

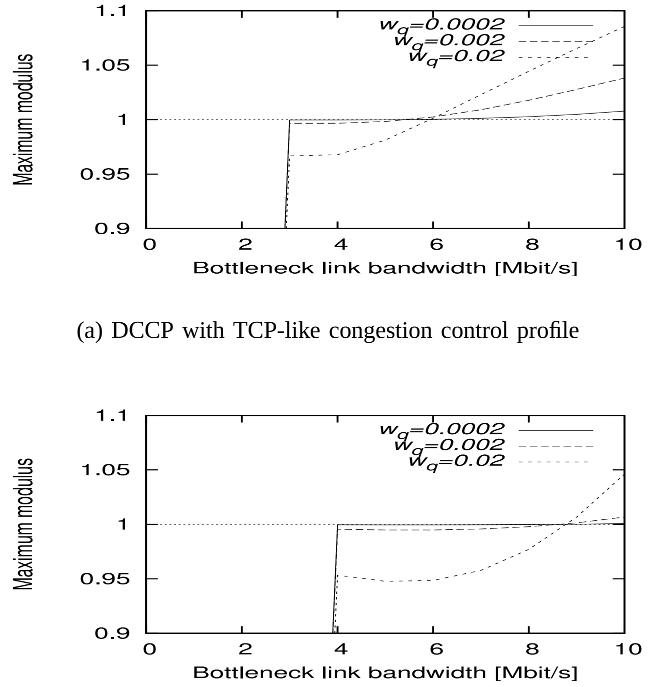
Finally, we investigate how the maximum modulus of the state transition matrix of DCCP/RED affects the transient state performance of DCCP/RED. Figure 5 shows the evolution of the average queue length $\overline{q}(k)$ of RED. Figure 5(a) shows results for DCCP with the TCP-like congestion control profile. Figure 5(b) shows for DCCP with the TFRC congestion control profile.

Furthermore, the average queue length $\overline{q}^*$, the maximum modulus λ of the state transition matrix of DCCP/RED, ramp-up time, overshoot and settling time are shown in Tab. 3. In our experiments, ramp-up time is defined as the time required for the average queue length of RED to reach 95% of the equilibrium value. Overshoot is defined as the maximum difference of the average queue length of RED from the equilibrium value. Settling time is defined as the time required for the average queue length of RED to be settled within 5% of the equilibrium value. The weight w_q of the exponential weighted moving average of RED is configured to 0.0002, 0.002 and 0.02. Moreover, the number of DCCP connections is $N = 1$, the bottleneck link bandwidth is $\mu = 4$ [Mbit/s], and the two-way propagation delay of DCCP is $\tau = 10$ [ms].

These results show that the ramp-up time and the settling time become small as the weight w_q of the exponential weighted moving average of RED becomes large. Moreover, comparison of DCCP with the TCP-like congestion control profile and DCCP with the TFRC congestion control profile indicates that each congestion control profile shows different characteristics regarding the overshoot. Namely, the overshoot of DCCP with the TCP-like congestion control profile becomes small as the weight w_q of the exponential weighted moving average becomes large. On the contrary, overshoot of DCCP with the TFRC congestion control profile becomes large as w_q becomes large.

7. RED-IQI (RED with Immediate Queue Information)

In Section 6, in the system where DCCP connections and RED routers are interconnected, we have shown that the settling time becomes large as the weight w_q of the exponential weighted moving average becomes small.



(a) DCCP with TCP-like congestion control profile

(b) DCCP with TFRC congestion control profile

Fig. 4: DCCP/RED transient state performance (maximum modulus of the state transition matrix)

TABLE 3

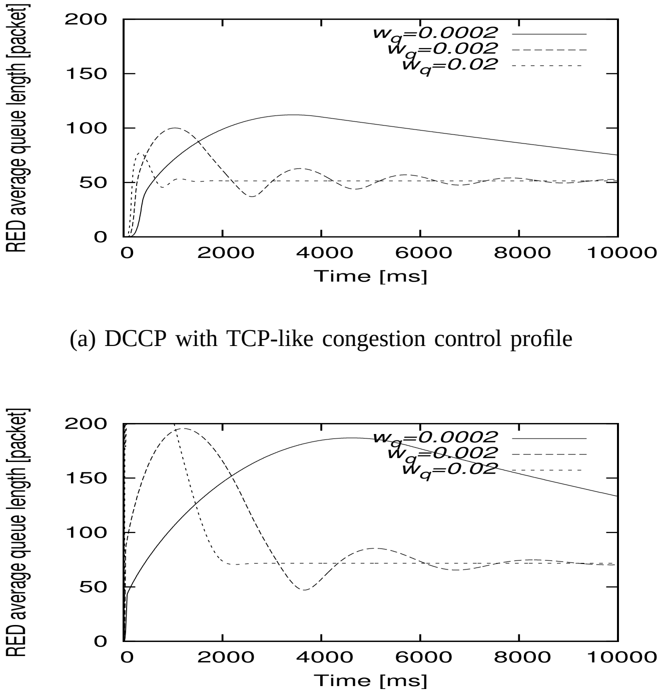
DCCP/RED TRANSIENT STATE PERFORMANCE INDICES

wq	profile	$\overline{q}^*$	λ	ramp-up time [ms]	overshoot [packet]	settling time [ms]
0.0002	CCID2	51.443	0.9996	560	27.846	36140
0.002	CCID2	51.443	0.9967	270	24.996	7960
0.02	CCID2	51.443	0.9678	180	17.217	920
0.0002	CCID3	71.724	0.9995	380	44.189	35320
0.002	CCID3	71.724	0.9954	40	45.408	7200
0.02	CCID3	71.724	0.9533	30	54.653	1960

The packet loss probability p_b of the RED router is determined by the liner function of $(\overline{q} - min_{th})/(max_{th} - min_{th})$ (Eq.9). We call $(\overline{q} - min_{th})/(max_{th} - min_{th})$ *queue occupancy*. Use of this function is determined without sufficiently taking account of the steady state performance and the transient state performance of RED. It is known that when the concave function is used as the function that determines the packet loss probability p_b of the RED router, the transient state performance and the robustness of RED improve [20].

Therefore, in this section, to improve the stability and transient state performance of the system where DCCP connections and RED routers are interconnected, we propose a RED-IQI (RED with Immediate Queue Information) by adding the following changes to RED.

First, we change the calculation method of the average queue length of RED. In RED-IQI, the weight of the exponential weighted moving average is set to $w_q = 1$. Thereby, the feedback delay of DCCP/RED-IQI becomes small, and the stability and the transient state performance are expected to improve. However, by configuring to $w_q = 1$, the packet loss probability of RED-IQI may sensitively fluctuate according to temporary variation of the network state. However, since the AIMD congestion control is used in the TCP-like congestion control profile, it is thought that the variation of the packet loss probability causes little performance degradation. On the other hand, since the TFRC congestion control profile



(a) DCCP with TCP-like congestion control profile

(b) DCCP with TFRC congestion control profile

Fig. 5: DCCP/RED transient state performance (evolution of RED average queue length)

smooths the packet loss event rate [28], it is thought that the variation of the packet loss probability is also causes little performance degradation.

Next, we change the function that determines the packet loss probability of RED. RED determines the packet loss probability using the linear function to the queue occupancy. In RED-IQI, we change this function to a concave function. Specifically, we change the function that determines the packet loss probability p_b to

$$p_b \quad = \quad max_p \, \mathcal{G}_\phi \left(\frac{\overline{q} - min_{th}}{max_{th} - min_{th}} \right), \qquad (15)$$

where $\mathcal{G}_\phi(x)$ is defined as

$$\mathcal{G}_\phi(x) \quad = \quad \left(1 - \sqrt{1-x^2}\right)^\phi. \qquad (16)$$

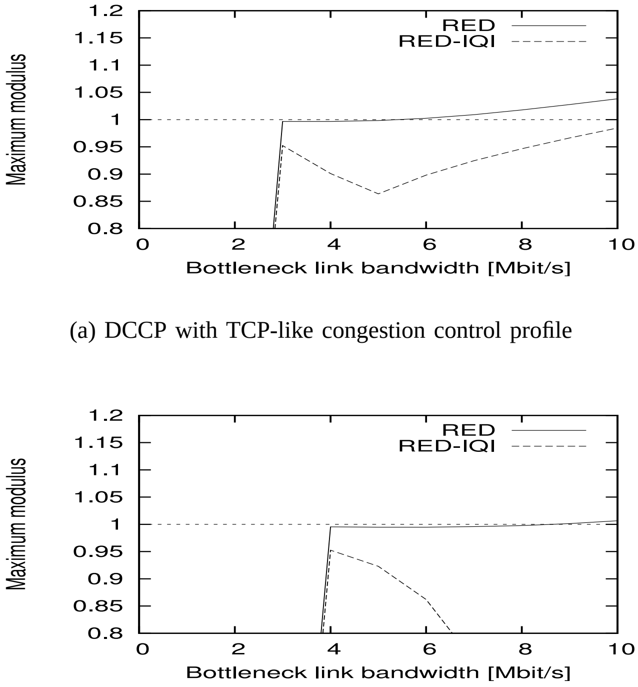
$\phi(> 0)$ is a parameter determining the concavity. In order for $\mathcal{G}_\phi$ to be concave,

$$\frac{d^2\mathcal{G}_\phi(x)}{dx^2} \quad = \quad \frac{1}{(1-x^2)\,\frac{3}{2}} \left\{ \phi \left(1 - \sqrt{1-x^2}\right)^{\phi-2} \times \left(1 + \sqrt{1-x^2}\, \left((\phi-1)\, x^2 - 1\right)\right) \right\} \geq 0$$

must be satisfied. By solving the above inequality for ϕ, we have

$$\phi \quad \geq \quad \lim_{x \to 0} \frac{-1 + \sqrt{1-x^2} + x^2\,\sqrt{1-x^2}}{x^2\,\sqrt{1-x^2}} = \frac{1}{2}. (17)$$

In what follows, by presenting several numerical examples, we show quantitatively how the transient state performance of DCCP/RED-IQI changes with the bandwidth and the propagation delay of the network. First, we focus on the transient state performance of DCCP/RED-IQI. Figure 6 shows the maximum modulus of the state transition matrix ($\mathbf{A}$ or $\mathbf{A}\,\mathbf{B}^{n-1}$) of DCCP/RED-IQI for different settings of the bottleneck link bandwidth. Figure. 6(a) shows results for DCCP with the TCP-like congestion control profile (Eq. (3)). Figure. 6(b) shows results for DCCP with the TFRC congestion control profile (Eq. (4)). For comparison purposes, the maximum modulus of the state transition matrix of DCCP/RED is also shown in the figure. Here, the weight w_q of the exponential weighted moving average of RED is configured to 0.002. Moreover, the number of DCCP connections is $N = 1$, and the two-way propagation delay of DCCP connection is $\tau = 10$ [ms].



(a) DCCP with TCP-like congestion control profile

(b) DCCP with TFRC congestion control profile

Fig. 6: DCCP/RED transient state performance (maximum modulus of the state transition matrix)

It can be found that the maximum modulus of DCCP/RED-IQI increases as the bottleneck link bandwidth increases from this figure. Moreover, by comparing the maximum modulus of DCCP/RED-IQI with that of DCCP/RED, it can be found that the value of DCCP/RED-IQI is smaller than that of DCCP/RED. This means that DCCP/RED-IQI operates more stably than DCCP/RED.

Next, we show the evolution of the average queue length $\overline{q}(k)$ of RED-IQI in Fig. 7. Furthermore, the average queue length $\overline{q}^*$, the maximum modulus λ of the state transition matrix, ramp-up time, overshoot and settling time of DCCP/RED-IQI are shown in Tab. 4. For comparison purposes, the average queue length $\overline{q}^*$, maximum modulus λ of the state transition matrix, ramp-up time, and overshoot and settling time of DCCP/RED are also shown in Tab. 4. Here, the weight w_q of the exponential weighted moving average of RED is configured to 0.002. The number of DCCP connections is $N = 1$, the bottleneck link bandwidth is $\mu = 4$ [Mbit/s], and the two-way propagation delay of DCCP is $\tau = 10$ [ms]. Figure 7(a) shows results for DCCP with the TCP-like congestion control profile. Figure 7(b) shows results for DCCP with the TFRC congestion control profile. These results show that the overshoot and the settling time of DCCP/RED-IQI become smaller and the ramp-up time of DCCP/RED-IQI becomes larger than those of DCCP/RED.

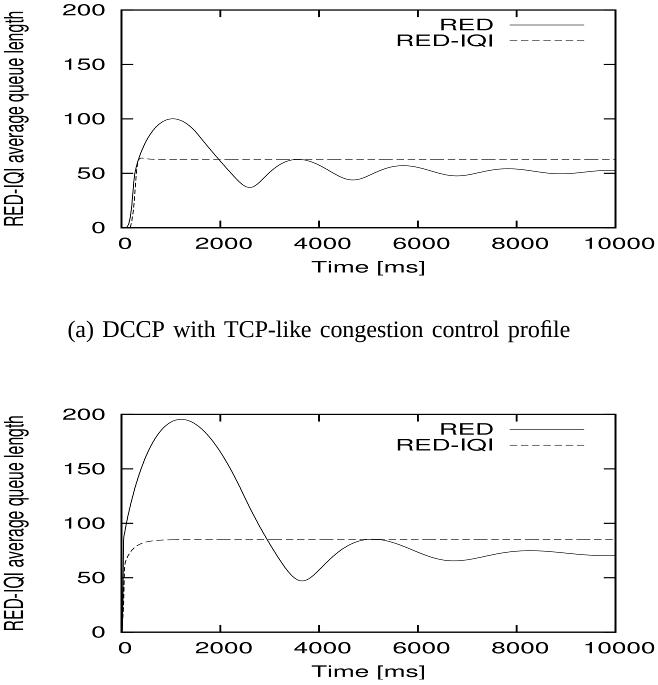
8. Conclusion and Future Work

In this paper, we have modeled DCCP congestion control mechanism and RED as independent discrete-time systems, and have modeled the entire network as a feedback system by interconnecting DCCP connections and RED routers. We have analyzed the steady state and transient state performance of DCCP/RED. We have derived the packet transmission rate of DCCP connections, the packet transmission rate, the packet loss probability, and the average queue length of the RED router in steady state. We have also derived the parameter region where DCCP/RED operates stably by linearizing DCCP/RED model around its equilibrium point. Furthermore, we have evaluated the transient state performance of DCCP/RED in terms of ramp-up time, overshoot, and settling time. Consequently, we have shown that the stability and the transient state performance of DCCP/RED degrade when the weight

TABLE 4

DCCP/RED AND DCCP/RED-IQI TRANSIENT STATE PERFORMANCE INDICES

	profile	$\overline{q}^*$	λ	ramp-up time [ms]	overshoot [packet]	settling time [ms]
RED	CCID2	51.443	0.9996	260	25.00	36140
RED-IQI	CCID2	62.715	0.9011	340	1.31	340
RED	CCID3	71.724	0.9995	40	45.41	35320
RED-IQI	CCID3	85.057	0.9525	380	0	380



(a) DCCP with TCP-like congestion control profile

(b) DCCP with TFRC congestion control profile

Fig. 7: DCCP/RED-IQI transient state performance (the average queue evolution of RED-IQI)

of the exponential weighted moving average is small. By adding changes to the function with which RED determines the packet loss probability, we propose RED-IQI. We have shown that RED-IQI significantly improves the transient state performance such as the maximum modulus, the overshoot and the settling time compared with RED. As future work, it would be interesting to analyze large-scale networks by applying the analytic approach proposed in [26] to the DCCP/RED model derived in this paper.

References

[1] E. Kohler, M. Handley, and S. Floyd, “Designing DCCP: Congestion control without reliability,” tech. rep., ICSI CENTER for Internet Research, 2003.

[2] J. Postel, “User datagram protocol,” *Request for Comments (RFC) 768*, Aug. 1980.

[3] M. Allman, V. Paxson, and W. R. Stevens, “TCP congestion control,” *Request for Comments (RFC) 2581*, Apr. 1999.

[4] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the Internet,” *IEEE Transactions on Networking*, vol. 7, pp. 458–472, May 1999.

[5] S. Floyd, M. Handley, J. Padhye, and J. Widmer, “Equation-based congestion control for unicast applications: the extended version,” tech. rep., International Computer Science Institute, Mar. 2000.

[6] E. Kohler, M. Handley, and S. Floyd, “Datagram congestion control protocol (DCCP),” *Internet Draft* `<draft-ietf-dccp-spec-11.txt>`, Mar. 2005.

[7] E. Kohler, M. Handley, and S. Floyd, “Profile for DCCP congestion control ID 2: TCP-like congestion control,” *Internet Draft* `<draft-ietf-dccp-ccid2-10.txt>`, Mar. 2005.

[8] E. Kohler, M. Handley, and S. Floyd, “Profile for DCCP congestion control ID 3: TFRC congestion control,” *Internet Draft* `<draft-ietf-dccp-ccid3-11.txt>`, Mar. 2005.

[9] B. Braden *et al.*, “Recommendations on queue management and

congestion avoidance in the Internet," *Request for Comments (RFC) 2309*, Apr. 1998.

[10] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[11] Y. Zhang and L. Qiu, "Understanding the end-to-end performance impact of RED in a heterogeneous environment," *Cornell CS Technical Report 2000-1802*, July 2000.

[12] S. Takeuchi, H. Koga, K. Iida, Y. Kadobayashi, and S. Yamaguchi, "Performance evaluations of DCCP for bursty traffic in real-time applications," in *Proceedings of 2005 International Symposium on Applications and the Internet (SAINT2005)*, pp. 142–149, 2005.

[13] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proceedings of IEEE INFOCOM 2000*, pp. 1435–1444, Mar. 2000.

[14] H. Ohsaki and M. Murata, "Steady state analysis of the RED gateway: stability, transient behavior, and parameter setting," *IEICE Transactions on Communications*, vol. E85-B, pp. 107–115, Jan. 2002.

[15] M. Kisimoto, H. Ohsaki, and M. Murata, "Analyzing the impact of TCP connections variation on transient behavior of RED gateway," in *Proceedings of the 16th International Conference on Information Networking (ICOIN-16)*, pp. 10A2.1–10A2.12, Jan. 2002.

[16] M. Kisimoto, H. Ohsaki, and M. Murata, "On transient behavior analysis of random early detection gateway using a control theoretic approach," in *Proceedings of the IEEE Control Systems Society Conference on Control Applications (CCA/CACSD 2002)*, pp. 1144–1149, Sept. 2002.

[17] C. V. Hollot, V. Misra, D. F. Towsley, and W. Gong, "A control theoretic analysis of RED," in *Proceedings of IEEE INFOCOM 2000*, pp. 1510–1519, Mar. 2001.

[18] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE INFOCOM 2001*, pp. 1726–1734, 2001.

[19] P. Kuusela, P. Lassila, and J. Virtamo, "Stability of TCP-RED congestion control," in *Proceedings of ITC-17*, pp. 655–666, Dec. 2001.

[20] H. Ohsaki and M. Murata, "On packet marking function of active queue management mechanism: Should it be linear, concave, or convex?," in *Proceedings of SPIE's International Symposium on the Convergence of Information Technologies and Communications (ITCom 2004)*, Oct. 2004.

[21] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," in *Proceedings of NOSSDAV'99*, 1999.

[22] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker, "Dynamic behavior of slowly-responsive congestion control algorithms," in *Proceedings of ACM SIGCOMM*, pp. 263–274, Aug. 2001.

[23] Y. R. Yang, M. S. Kim, and S. S. Lam, "Transient behaviors of TCP-friendly congestion control protocols," *The International Journal of Computer and Telecommunications Networking*, vol. 41, pp. 193–210, Feb. 2003.

[24] K. Ramakrishnan, S. Floyd, and D. Black, "The addition of explicit congestion notification (ECN) to IP," *Request for Comments (RFC) 3168*, Sept. 2001.

[25] N. Spring, D. Wetherall, and D. Ely, "Robust explicit congestion notification (ECN) signaling with nonces," *Request for Comments (RFC) 3540*, June 2003.

[26] H. Ohsaki, J. Ujiie, and M. Imase, "On scalable modeling of TCP congestion control mechanism for large-scale IP networks," in *Proceedings of the 2005 International Symposium on Applications and the Internet (SAINT 2005)*, pp. 361–369, Feb. 2005.

[27] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proceedings of ACM SIGCOMM '98*, pp. 303–314, Sept. 1998.

[28] M. Handly, S. Floyd, J. Padhye, and J. Widmer, "TCP friendly rate control (TFRC): protocol specification," *Request for Comments (RFC) 3448*, Jan. 2003.

[29] N. S. Nise, *Control Systems Engineering*. New York: John Wiley & Sons, 4th ed., Aug. 2003.

THE AUTHOR

Hiroyuki Hisamatsu received the M. E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 2003. He is currently a graduate student at the Graduate School of Information Science and Technology, Osaka University. His research work is in the area of performance evaluation of TCP/IP networks. He is a student member of IEEE and IEICE.

Hiroyuki Ohsaki received the M. E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 1995. He also received the Ph. D. degree from Osaka University, Osaka, Japan, in 1997. He is currently an associate professor at Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of traffic management in high-speed networks. He is a member of IEEE and Institute of Electronics, Information, and Computer Engineers of Japan (IEICE). His e-mail address is oosaki@ist.osaka-u.ac.jp

Masayuki Murata received the M.E. and D.E. degrees from Osaka University, Japan, in 1984 and 1988, respectively. In 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. He was an Assistant Professor of Computation Center, Osaka University from 1987 to 1989, an Assistant Professor of Faculty of Engineering Science from 1989 to 1992, an Associate Professor of Graduate School of Engineering Science from 1992 to 1999. From 1999, he has been a Professor of Osaka University. He moved to Graduate School of Information Science and Technology in 2004. He has more than three hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, The Internet Society, IEICE and IPSJ.