IEICE TRANS. COMMUN., VOL.E91–B, NO.9 SEPTEMBER 2008

PAPER *IEICE/IEEE Joint Special Section on Autonomous Decentralized Systems Theories and Application Deployments*

Parameter Design for Diffusion-Type Autonomous Decentralized Flow Control

Chisa TAKANO[†a)], **Keita SUGIYAMA**[††], ***and*** **Masaki AIDA**[††], ***Members***

SUMMARY We have previously proposed a diffusion-type flow control mechanism as a solution for severely time-sensitive flow control required for high-speed networks. In this mechanism, each node in a network manages its local traffic flow using the basis of only the local information directly available to it, by using predetermined rules. In addition, the implementation of decision-making at each node can lead to optimal performance for the whole network. Our previous studies show that our flow control mechanism with certain parameter settings works well in high-speed networks. However, to apply this mechanism to actual networks, it is necessary to clarify how to design a parameter in our control mechanism. In this paper, we investigate the range of the parameter and derive its optimal value enabling the diffusion-type flow control to work effectively.

key words: *flow control, autonomous decentralized control, diffusion equation, high-speed networks*

1. Introduction

A high-speed network has been constructed with rapid spread of the Internet and expanding demand for services in recent years. In such a high-speed network, the fast and flexible network control is required so that various application users can be provided with stable Quality of Service (QoS) and the limited network resource can be used effectively.

In a high-speed network, it is impossible to implement time-sensitive control based on collecting global information about the whole network because the state of a node varies rapidly in accordance with its processing speed although the propagation delay is constant. This is because the propagation delay is constrained by the speed of light and is the same as in slower networks although the processing speed of each node is high in high-speed networks. If we allow sufficient time to collect network-wide information, the data so gathered is too old to use for time-sensitive control. In this sense, each node in a high-speed network is isolated from up-to-date information about the state of other nodes or that of the overall network.

This paper focuses on a flow control mechanism for high-speed networks. From the above considerations, the technique used for our flow control method should satisfy the following requirements: (i) it must be possible to collect the information required for the control method, and (ii) the control should take effect immediately.

Manuscript received December 28, 2007.
Manuscript revised April 18, 2008.

[†]The author is with the Graduate School of Information Sciences, Hiroshima City University, Hiroshima-shi, 731-3194 Japan.
[††]The authors are with the Graduate School of System Design, Tokyo Metropolitan University, Hino-shi, 191-0065 Japan.
a) E-mail: takano@hiroshima-cu.ac.jp

DOI: 10.1093/ietcom/e91–b.9.2828

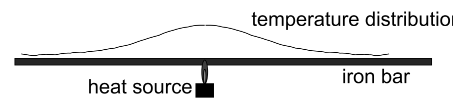


Fig. 1 Example of thermal diffusion phenomena.

There are many other papers reporting studies on flow control optimization in a framework of solving linear programs [1]–[5]. These studies assume the collection of global information about the network, but it is impossible to achieve such a centralized control mechanism in high-speed networks. In addition, solving these optimization problems requires enough time to be available for calculation, so it is difficult to apply these methods to decision-making on a very short timescale. Therefore, in a high-speed network, the principles adopted for time-sensitive control are inevitably those of autonomous decentralized systems.

For the time-sensitive autonomous decentralized flow control, we considered a new control mechanism, diffusion-type flow control (DFC) [6]–[9] in which the nodes in a network handle their local traffic flows themselves, based only on the local information directly available to them. In this mechanism, each node can immediately detect a change in the network state around the node and apply quick decision-making.

The most remarkable characteristic of DFC is that it provides a framework in which the implementation of the decision-making of each node leads to high and stable performance for the whole network. To explain the basis of this framework, we show the principle of our flow control model through the following analogy [8]. When we heat a point on a cold iron bar, the temperature distribution follows a normal distribution and heat spreads through the whole bar by diffusion (Fig. 1). In this process, the action in a minute segment of the iron bar is very simple: heat flows from the hotter side towards cooler side. The rate of heat flow is proportional to the temperature gradient. There is no communication between two distant segments of the iron bar. Although each segment acts autonomously, based on its local information, the temperature distribution of the whole iron bar exhibits orderly behavior. In DFC, each node controls its local packet flow, which is proportional to the difference between the number of packets in the node and that in an adjacent node. Thus, the distribution of the total number of packets in a node in the network becomes uniform over time. In this control mechanism, the state of the whole network is controlled indirectly through the autonomous action of each node.

Our previous studies show that our flow control mechanism with certain parameter settings works well in high-speed networks. However, to apply DFC to actual networks, it is necessary to clarify how to design parameters in our control mechanism. This is one of central issues to be solved for applying DFC to actual networks. In this paper, we investigate the range of the parameter and derive its optimal value enabling the DFC to work effectively. This paper is organized as follows. In Sect. 2, we show the aim of the DFC in layered structure of network control with respect to timescale. In Sect. 3, we briefly summarize the mechanism of DFC. In Sect. 4, we clarify how to design the parameter in our control mechanism. The validity of the parameter design is verified in Sect. 5 by using simulation studies. Finally, Sect. 6 shows conclusions.

2. Aim of Diffusion-Type Flow Control Mechanism

Network control mechanisms in networks can be categorized by the timescales on which an individual control takes effect and they form a layered structure with respect to the timescales. Individual control mechanisms work well on their appropriate timescales and they cooperate with each other. Figure 2 shows the comparison of different types of control according to such a classification. For example, the routing and signaling (e.g. session initiation protocol (SIP)) respectively fall into the long and medium timescales. Individual control mechanisms work well for their appropriate timescales and they cooperate with each other for supporting user applications. TCP is a typical decentralized flow control by end hosts [10], and is widely used for reliable communications in current networks. Window flow control such as TCP acts in the timescale of the round-trip time (RTT).

End-to-end or end-to-node control cannot be applied to decision-making on a timescale shorter than the RTT. Indeed, in low-speed networks, a control delay on the order of the RTT has a negligible effect on the network performance. However, in high-speed networks, the control delay greatly affects the network performance. This is because the RTT becomes large relative to the unit of time determined by node's processing speed, although the RTT is itself unchanged. This means that nodes in high-speed networks experience a larger RTT, and this causes an increase in the sensitivity to control delay. This is a significant reason that the performance of TCP should be sensitive with respect to network condition, and a lot of extensions of TCP have been proposed for high-bandwidth or long-distance connections [11]–[16].

Let us consider the network conditions that the bandwidth and/or the distance of communications become(s) larger. If the performance characteristics concerning packet loss of its bearer network are little affected by this condition, we can expect that TCP would not enter the congestion avoidance phase and that high network performance would be achieved. To achieve such a stabilization mechanism, it is necessary to implement a time-sensitive control mechanism in the bearer network and to make the network have robust performance characteristics. Since end-to-end or end-to-node control cannot be applied to decision-making on a timescale shorter than the RTT, it is inadequate to support decision-making on a very short timescale.

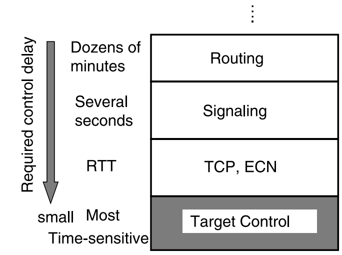


Fig. 2 Classification of various control mechanisms with respect to their effective timescales.

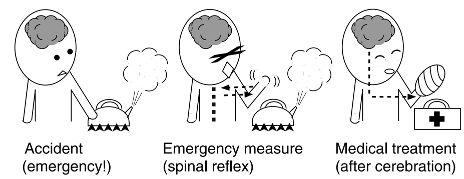


Fig. 3 Layered structure of behaviors with respect to timescales.

To achieve rapid control on a shorter timescale than the RTT, it is preferable to apply control by the nodes rather than by the end hosts. DFC is a node-by-node control and its target is a timescale shorter than the RTT. The aim of DFC is not direct guarantee of user QoS, but DFC serves stable network behaviors to other controls of the longer timescales. In this sense, DFC brings its ability in the background of other controls of the longer timescales. In other words, DFC enables other controls of the longer timescales to work well even if the speed of the networks becomes high.

Figure 3 shows a sequence of human behavior as an analogy. When an accident occurs, we immediately act to guard ourselves. This action is time-sensitive and is achieved by a spinal reflex. After that, essential treatment is performed through cerebration on a relatively long timescale. As well as the relation of the cerebration and spinal reflex, network control mechanisms of different timescales act on their own timescales and, as a result, they cooperate with each other to support user applications.

3. Preliminary

3.1 Diffusion-Type Flow Control Mechanism

In the case of Internet-based networks, to guarantee end-to-end QoS of a flow, the QoS-sensitive flow has a static route (*e.g.*, RSVP). Thus, we assume that a target flow has a static route. In addition, we assume all routers in the network can employ per-flow queueing for all the target flows†. All flows are in the same priority class and it is desirable that all active flows share the link bandwidth fairly.

In DFC, each node controls its local packet flow au-

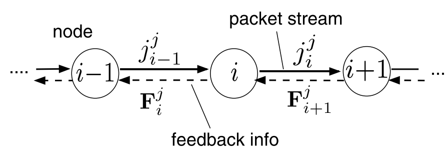


Fig. 4 Node interactions in our flow control model.

tonomously. Figure 4 shows the interactions between nodes (routers) in our flow control method, using a network model with a simple 1-dimensional configuration. All nodes have two incoming and two outgoing links, for a one-way packet stream and for feedback information, that is, node i ($i = 1, 2, \ldots$) transfers packets to node $i+1$, and node $i+1$ sends feedback information $\mathbf{F}_{i+1}$ to node i. For simplicity, we assume that packets have a fixed length in bits.

All nodes are capable of receiving feedback information from the adjacent downstream nodes, and sending it to the adjacent upstream nodes. Each node i can receive feedback information sent from the downstream node $i+1$ and can send feedback information about itself to the upstream node $i-1$.

When node i receives feedback information from downstream node $i+1$, it determines the transmission rate for packets to the downstream node $i+1$ using the received feedback information, and it adjusts its transmission rate towards the downstream node $i+1$.

Assume that there are M_i flows sharing the link between node i and $i+1$, and they are identified by j ($j = 1, 2, \ldots, M_i$). The framework for node behavior and flow control for flow j is summarized as follows:

- Each node i autonomously determines the transmission rate J_i^j for flow j on the basis of only the local information directly available to it, that is, the feedback information obtained from the downstream node $i+1$ and its own feedback information.
- The rule for determining the transmission rate is the same for all nodes.
- Each node i adjusts its transmission rate towards the downstream node $i+1$ to J_i^j.
  (If there are no packets for flow j in node i, the packet transmission rate is 0.)
- Each node i autonomously creates feedback information according to a predefined rule and sends it to the upstream node $i-1$. Feedback information is created periodically with a fixed interval τ_i.
- The rule for creating the feedback information is the same for all nodes.
- Packets and feedback information both experience the same propagation delay.

As mentioned above, the framework of our flow control model involves both autonomous decision-making by each node and interaction between the adjacent nodes. There is no centralized control mechanism in the network.

Next, we explain the details of DFC. The transmission rate $J_i^j(t)$ for flow j of node i at time t is determined by

$$J_i^j(t) = \max(0, \min(L_i^j(t), \tilde{J}_i^j(t))), \quad \text{and} \tag{1}$$

$$\tilde{J}_i^j(t) = r_i^j(t-d_i) - D_i\,(n_{i+1}^j(t-d_i) - n_i^j(t)), \tag{2}$$

where $L_i^j(t)$ denotes the available bandwidth for flow j of the link from node i to node $i+1$ at time t, $n_i^j(t)$ denotes the number of packets belonging to flow j in node i at time t, $r_i^j(t-d_i)$ is the notified rate for flow j by using feedback information from the downstream node $i+1$, and d_i denotes the propagation delay between nodes i and $i+1$.

The way to determine $L_i^j(t)$ is explained. Let the bandwidth of the link from node i to node $i+1$ be B_i, and $L_i^i(t)$, the available bandwidth for flow j is to assume that the bandwidth B_i is shared by flow with a weight $\tilde{J}_i^j(t)$ [17], that is,

$$L_i^j(t) = B_i\,\frac{\tilde{J}_i^j(t)}{\sum_{j=1}^{M_i} \tilde{J}_i^j(t)}. \tag{3}$$

This rule means that a flow with larger $\tilde{J}_i^j(t)$ can get a larger transmission rate and can transmit a larger volume of traffic to the downstream node. Thus, the transmission rates of other flows are regulated to be smaller.

In addition, $r_i^j(t-d_i)$ and $n_{i+1}^j(t-d_i)$ are reported from the downstream node $i+1$ as feedback information with propagation delay d_i. Parameter D_i is chosen to be inversely proportional to the propagation delay [7] as follows:

$$D_i = \frac{D}{d_i}, \tag{4}$$

where D (> 0), which is a positive constant, is the diffusion coefficient.

The feedback information for flow j created every fixed period τ_i by node i consists of the following two quantities:

$$\mathbf{F}_i^j(t) = (r_{i-1}^j(t), n_i^j(t)). \tag{5}$$

Node i reports this to the upstream node $i-1$ with a period of $\tau_i = d_{i-1}$. Here, the target transmission rate for flow j is determined as

$$r_{i-1}^j(t) = J_i^j(t). \tag{6}$$

Moreover, the transmission rate $J_i^j(t)$ for flow j in node i is renewed whenever feedback information arrives from the downstream node $i+1$ (with a period of $\tau_{i+1} = d_i$).

To enable an intuitive understanding, we briefly explain the physical meaning of DFC. We replace i with x and apply a continuous approximation. Then the propagation delay becomes $d_i \to 0$ for all i and the transmission rate (2) for flow j is expressed as

$$\tilde{J}^j(x,t) = r^j(x,t) - \kappa\,\frac{\partial n^j(x,t)}{\partial x}, \tag{7}$$

where κ is a diffusion coefficient corresponding to D in DFC.

[†]The assumption of per-flow queueing is for simplicity of explanations of DFC mechanism. The requirement of "per-flow" can be relaxed to "per input-port" of a router.

Since κ has an important meaning, a later section explains this parameter. The temporal evolution of the packet density $n^j(x, t)$ may be represented by a diffusion-type equation,

$$\frac{\partial n^j(x,t)}{\partial t} = -\frac{\partial r^j(x,t)}{\partial x} + \kappa \frac{\partial^2 n^j(x,t)}{\partial x^2}, \tag{8}$$

using the continuous equation

$$\frac{\partial n^j(x,t)}{\partial t} = -\frac{\partial \tilde{J}^j(x,t)}{\partial x}. \tag{9}$$

As explained in Sect. 1, our method aims to perform flow control using the analogy of diffusion. We can expect excess packets in a congested node to be distributed over the whole network and expect normal network conditions to be restored after some time.

In addition to the above framework, we consider the boundary condition of the rule for determining the transmission rate in the DFC.

Here we consider the situation where nodes and/or end hosts in other networks do not support the DFC mechanism. We call the nodes and/or end hosts that are connected directly to the ingress node in our network external nodes. We only assume that the external nodes have a traffic shaping function, that can adjust the transmission rate by queueing to the requested rate reported by the downstream node. That is, an external node 0 cannot calculate the transmission rate $J_0^j(t)$ for flow j using (2), but can adjust its transmission rate to $r_0^j(t - d_0)$ for flow j, which was reported by node 1.

We consider a rule for determining $r_0^j(t)$ as a boundary condition. Node 1 can calculate $J_0^j(t)$ if we assume that the number of packets stored in the external node is $i = 0$. The target rate $r_0^j(t)$ for flow j, reported by node 1, is created as $\tilde{J}_0^j(t)$ with the above assumption. That is,

$$r_0^j(t) := \tilde{J}_0^j(t + d_0) = J_1^j(t) - D_0\, n_1^j(t). \tag{10}$$

This quantity can be calculated just from information known to node 1.

It is worth to note that the packet rate of (2) is "non-work-conserving," in general. If each node acts as "work-conserving," many packets are concentrated at congested nodes and it causes packet losses. Although each node autonomously determines its packet rate, DFC nodes cooperate to work to avoid the concentration of packets. Notice that "non-work-conserving" does not mean the degradation of the utilization of networks. Even if a node transmits the packets at "work-conserving" rate, they are stored at the congested node but are not transferred to their destinations.

4. Parameter Design

4.1 Approach

In DFC, there is a important parameter, the diffusion coefficient D. The diffusion coefficient governs the speed of diffusion. In physical diffusion phenomenon, larger κ in (8) causes faster diffusion. If DFC model is completely corresponding to physical diffusion phenomenon, a large value of D is suitable for fast recovery from congestion. Unfortunately, DFC is not completely corresponding to physical diffusion and too large value of D in DFC prevents diffusion phenomenon in networks. Conversely, too small value of D causes very slow diffusion, and this means that stolid congestion recovery wastes much time.

In this section, we determine an appropriate value of D. Our approach to design a value of D is simple. We take a larger value of D in the range of values in which diffusion can occur in networks.

In the subsequent subsections, we discuss the following issues in order to determine the appropriate range of D.

- To determine the range of κ in which diffusion effect can occur for discrete space and discrete time.
- To determine the range of D in which diffusion can occur in networks, under the situation that all the links in networks have same length.
- To verify the above range of D is applicable to general situations that the length of links in networks different, in general.

For the first issue, we introduce the difference equation, which is corresponding to the diffusion equation with the parameter κ, and consider the range of κ. Next, for the second issue, we consider the close relation between the range of κ and the range of D. For the last issue, deriving the dimensionless property of D, we show the range of D is independent of the link length.

4.2 Range of Diffusion Coefficient under Homogeneous Network Configurations

The partial differential Eq. (8) describes temporal evolution of packet density in continuous approximation of networks. The first term on the right-hand side in (8) describes a stationary packet flow, and this is not concerned to diffusion. The second term is essential in diffusion. Thus, we consider the following partial differential equation,

$$\frac{\partial n^j(x,t)}{\partial t} = \kappa \frac{\partial^2 n^j(x,t)}{\partial x^2}, \tag{11}$$

where this is the ordinary diffusion equation.

Of course, the structure of networks are not continuous. In addition, timing of control actions is not continuous. The behavior of DFC is described by a difference equation rather than the differential equation. In other words, DFC make networks solve a difference equation with discrete space x and discrete time t.

To introduce a discrete space reflecting the network structure, we divide the continuous 1-dimensional space into a length of Δx. In addition, for simplicity, we assume all the links in networks have same length. Since we set $d_i = \tau_{i+1}$, this means the interval of DFC's actions is the same for all node, and we denote it as Δt. The difference equation corresponding to (11) is as follows:

$$\frac{n^j(x, t+\Delta t) - n^j(x, t)}{\Delta t} = \kappa \frac{n^j(x+\Delta x,\ t) - 2\,n^j(x,\ t) + n^j(x-\Delta x,\ t)}{(\Delta x)^2}. \quad (12)$$

If the solution of (12) exhibits similar behavior to that of (11), DFC appropriately works and diffusion of packet density occurs. Our issue is to find appropriate value of D in which the solution of (12) exhibits diffusion phenomenon.

Let node position in 1-dimensional configuration be x_k ($x_{k+1} - x_k = \Delta x$; $k = 0, 1, \ldots, S$), and time of DFC's action be t_ℓ ($t_{\ell+1} - t_\ell = \Delta t$; $\ell = 0, 1, \ldots, T$). We take the boundary condition,

$$n^j(x_0,\ t) = n^j(x_S,\ t) = 0. \quad (13)$$

If behavior of $n^j(x_k,\ t_\ell)$ exhibits a diffusion effect with time,

$$\lim_{\ell\to\infty} n^j(x_k,\ t_\ell) = 0, \quad (14)$$

for all k.

In general, $n^j(x_k,\ t_\ell)$ satisfying (13) is represented as the following Fourier series,

$$n^j(x_k,\ t_\ell) = \sum_{m=0}^{\infty} n^j_m(x_k,\ t_\ell), \quad (15)$$

and

$$n^j_m(x_k,\ t_\ell) = A^j_{m,\ell} \sin\left(\frac{k\,m\,\pi}{S}\right), \quad (16)$$

where $A^j_{m,\ell}$ is a time-dependent coefficient. If (14) is valid in any cases,

$$\lim_{\ell\to\infty} n^j_m(x_k,\ t_\ell) = 0 \quad (17)$$

for all non-negative integers m. By substituting (16) into (12), we have

$$A^j_{m,\ell+1} = A^j_{m,\ell} \left(1 - \frac{4\,\kappa\,\Delta t}{(\Delta x)^2} \sin^2 \frac{m\,\pi}{2\,S}\right), \quad (18)$$

and then, $A^j_{m,\ell}$ can be obtained from $A_{m,0}$ as

$$A^j_{m,\ell} = A^j_{m,0} \left(1 - \frac{4\,\kappa\,\Delta t}{(\Delta x)^2} \sin^2 \frac{m\,\pi}{2\,S}\right)^\ell. \quad (19)$$

Therefore,

$$n^j_m(x_k,\ t_\ell) = A^j_{m,0} \left(1 - \frac{4\,\kappa\,\Delta t}{(\Delta x)^2} \sin^2 \frac{m\,\pi}{2\,S}\right)^\ell \sin\left(\frac{k\,m\,\pi}{S}\right). \quad (20)$$

From (14), κ should satisfy

$$\left|1 - \frac{4\,\kappa\,\Delta t}{(\Delta x)^2} \sin^2 \frac{m\,\pi}{2\,S}\right| < 1 \quad (21)$$

and we obtain the range of the diffusion coefficient κ,

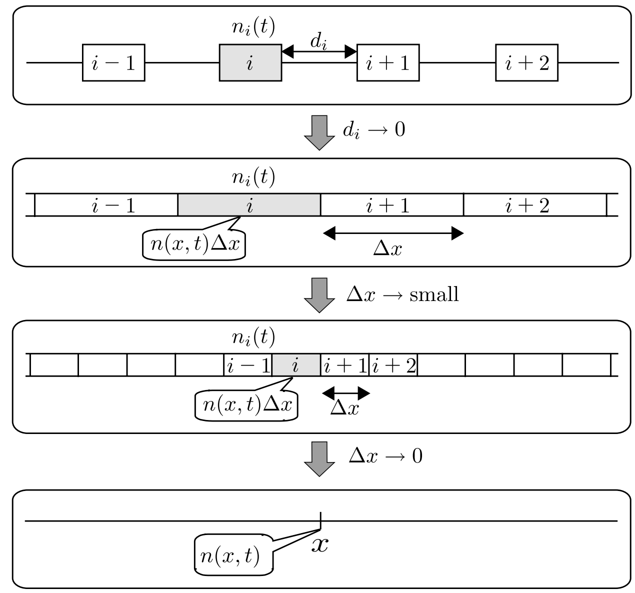


Fig. 5 Correspondence between discrete and continuous network models.

$$0 < \kappa < \frac{1}{2} \frac{(\Delta x)^2}{\Delta t}. \quad (22)$$

Note that the reason that the inequality (21) does not include the equality comes from conditions (14) and (17). If both conditions are not equal to 0 but finite, the left-hand side of inequality (21) is less than or equal to 1. This constraint of κ is the same as the constraint that appears in solving (11) by discrete space-time computations. In order to obtain the range of the diffusion coefficient κ, it is necessary to consider the relationship between κ and D. It is shown in the next subsection.

4.3 Parameter Design of DFC and Dimensional Analysis

In this subsection, we consider the actual value of D by dimensional analysis. We consider the value of the diffusion coefficient expressed in the ordinary MKS (Meter/Kilogram/Second) unit system.

In order to consider the relationship between κ and D, let us reconsider the meaning of related parameters and refine a continuous approximation of the network. Although we assumed $d_i = \tau_{i+1}$, which means the link delay is equal to the interval of DFC's actions, they are different notion originally. So, we regard them, in this subsection, as different (independent) quantities. In accordance with the modification, we redefine

$$D_i = \frac{D}{\tau_{i+1}}, \quad (23)$$

instead of (4).

The relationship between the discrete space and the continuous space are shown in Fig. 5. First, we take the limit of the distance between the adjacent nodes as $d_i \to 0$. The second panel in Fig. 5 shows this situation. Here, the interval of DFC's actions Δt is unchanged. Next, taking the limit of Δx to be 0, we obtain a continuous approximation of the 1-dimensional network. Note that Δt should approach 0 concurrently with $\Delta x \to 0$. The relationship between the packet density $n(x,t)$ and the number of packet stored in a node $n_i(t)$ is also shown in Fig. 5.

After taking the limit as $d_i \to 0$, we compare the second term on the right hand side of (2) with that of (7),

$$\kappa \frac{\partial n^j(x,t)}{\partial x} = D \lim_{\Delta t \to 0} \frac{(n^j_{i+1}(t) - n^j_i(t))}{\Delta t}, \tag{24}$$

where we use (23) and $\Delta t = \tau_{i+1}$. These quantities in both sides in (24) are parts of packet rate and have a dimension of packet/s. Note that $n^j(x,t)$ denotes the packet density and its dimension is packet/m. From Fig. 5,

$$n^j(x+\Delta x, t) - n^j(x,t) = \frac{n^j_{i+1}(t) - n^j_i(t)}{\Delta x},$$

where $i = \lceil x/\Delta x \rceil + 1$.

Therefore, we have

$$\begin{aligned} \kappa \frac{\partial n^j(x,t)}{\partial x} &= \kappa \lim_{\Delta x \to 0} \frac{n^j(x+\Delta x,t) - n^j(x,t)}{\Delta x} \\ &= \kappa \lim_{\Delta x \to 0} \frac{n^j_{i+1}(t) - n^j_i(t)}{(\Delta x)^2} \\ &= \kappa \lim_{\Delta t \to 0} \frac{\Delta t}{(\Delta x)^2} \frac{n^j_{i+1}(t) - n^j_i(t)}{\Delta t}. \end{aligned} \tag{25}$$

In the last equality, Δx should approach 0 concurrently with Δt in order to remain $\Delta t/(\Delta x)^2$ a constant. Since $\Delta t = d_i$ and by comparing (25) with (24), we obtain

$$D = \kappa \frac{\Delta t}{(\Delta x)^2}. \tag{26}$$

Since κ has the dimension of [m^2/sec] from (22), D is a dimensionless parameter. From this result and (22), we have the range of the diffusion parameter D as

$$0 < D < \frac{1}{2}. \tag{27}$$

We assumed homogeneous network configurations in the above considerations. Next, we consider the condition that links have different length, in general. Since the diffusion parameter D is dimensionless, we can expect that D is invariant even if the length of link is changed. If links have different length, the intervals of DFC's actions are different. Let the interval of DFC's action for node k be Δt_k. Then, (22) can be generalized as

$$0 < \kappa(k) < \frac{1}{2} \frac{(\Delta x)^2}{\Delta t_k}, \tag{28}$$

and the diffusion coefficient $\kappa(k)$ depends on node position k, that is, link length. On the other hand, (26) can be generalized as

$$D = \kappa(k) \frac{\Delta t_k}{(\Delta x)^2}, \tag{29}$$

therefore, (27) is valid even in the situation where links have different length.

Our design policy of D is to take a larger value of D in the range (27).

5. Simulation Studies

In this section, we show simulation studies about the performance of DFC with different values of the diffusion coefficient D in order to verify the range (27) and our design policy of D. Simulations were made by using ns2 simulator [18]. We extended the simulation tool ns2 capability with the function of DFC.

5.1 Simulation

The purpose of our simulation studies is to verify that the range (27) derived from theoretical analysis is applicable also to actual networks. To this end, it is important to evaluate the response of a single congestion event. The reasons are as follows. DFC is based on a linear differential equation (11) or more specifically a linear difference equation (12). In such linear systems, in general, the response with respect to an impulse input is important and it is frequently called the Green function. If we have the Green function, we can obtain general solutions by superposing the Green function. In our case, the impulse input corresponds to a single congestion event. If the range (27) is applicable to simulation scenarios having a single congestion event, the range (27) is applicable also to other simulation scenarios under more complex and general traffic conditions. Thus, we evaluate the range of D by using the following simple simulation model.

Figure 6 shows our network model with 30 nodes, which is used in the simulations. Although this 1-dimensional model looks simple, it represents a part of a network and describes a path of the target end-to-end flow extracted from the whole network. The capacity of buffer at each node is 1800 packets. A packet has a fixed length of 1500 Bytes and the link bandwidth is 1,000,000 packets/s. This means the link bandwidth is 12 Gbps. Both flows have greedy traffic, that is, the rate of each flow is as large as possible.

This paper uses inhomogeneous network configurations, that is, the length of links are different. The length of link is described by the propagation delay, and it obeys a probability distribution with the mean of 0.1 ms. Since the speed of light in optical fiber is about 2/3 of the speed in the vacuum (300,000 km/s), this model assumes the link length of about 20 km. The propagation delay of each link is determined in advance. The delay consists of a fixed component of 0.05 ms and the variable component that obeys an exponential distribution with the mean of 0.05 ms.

The simulation scenario is as follows. There are two TCP Reno flows. The target flow is between node 1 and node 30, while the background traffic flows between node 15 and node 30. The maximum TCP window size and the initial

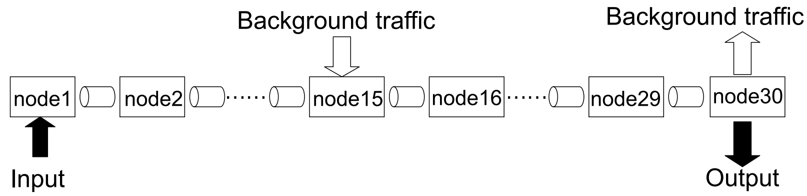


Fig. 6 Simulation model.

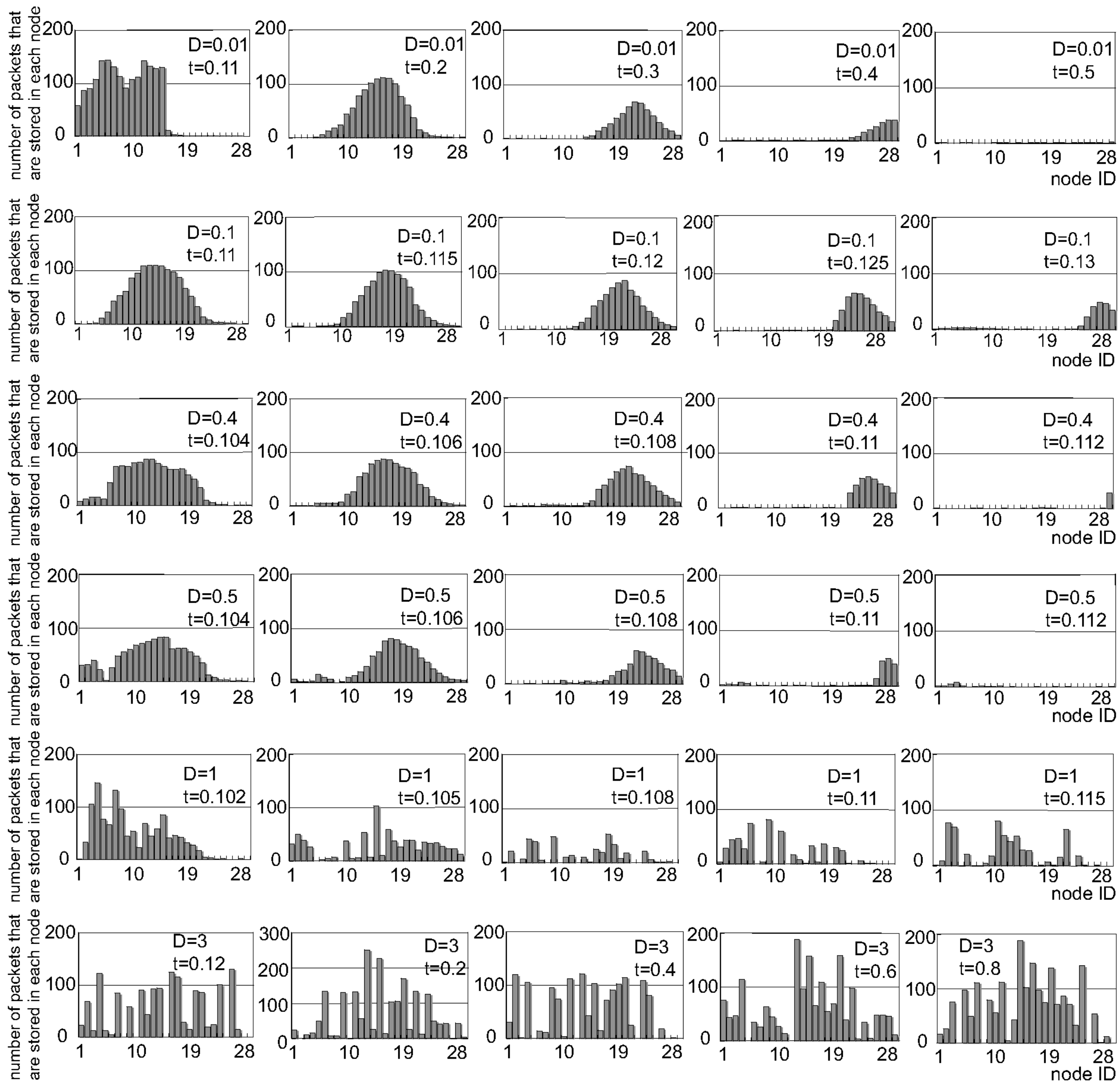


Fig. 7 Temporal evolution of distribution of packets stored in each node.

TCP window size of both flows are 10,000 and it is chosen as sufficiently larger than the bandwidth-delay product of RTT.

The reasons why we combine TCP and DFC in our simulation model are as follows. Since TCP performance is strongly influenced by the network QoS, we can clarify the stability of network performance by DFC by observing TCP performance. We can detect decreases of the throughput by a reduction of the TCP window size when packet loss occurs. However, we use only a part of TCP functions. We assume that the ingress node has the shaping mechanism in DFC. This DFC shaping controls the volume of traffic between the application layer and the TCP layer. The excessive traffic cannot enter the network even if TCP window

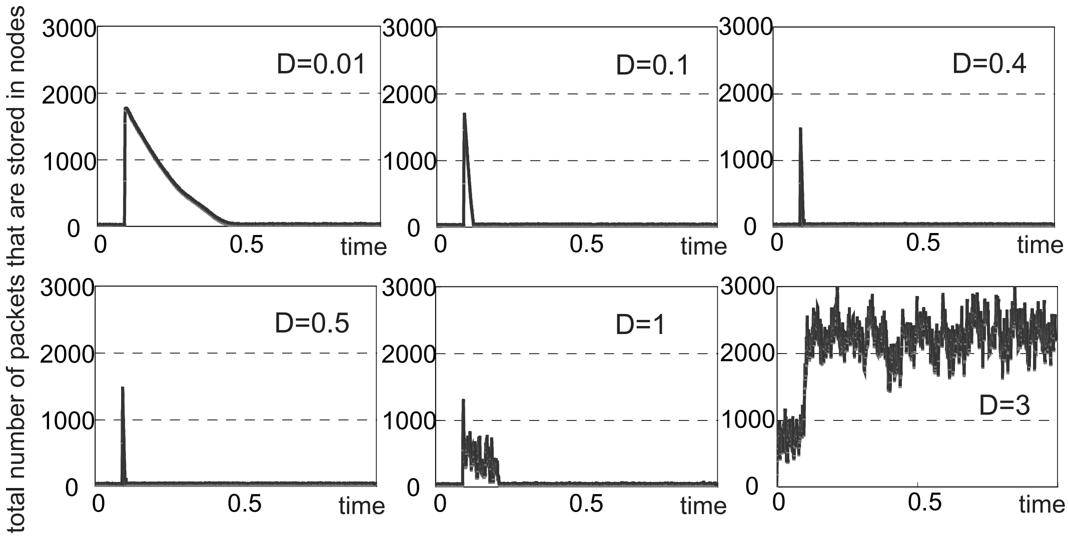


Fig. 8 Temporal evolution of the total number of packets that are stored in nodes.

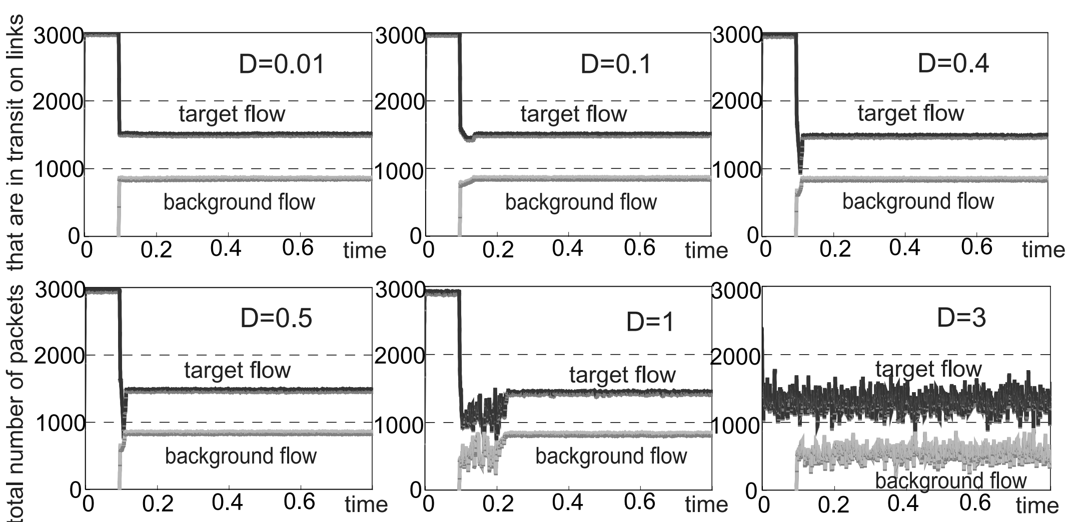


Fig. 9 Temporal evolution of the total number of packets that are in transit on links.

size is very large. If the number of packets stored in nodes is smoothly distributed over the network by DFC, there will be no packet loss in setting the appropriate buffer size. In our simulation, we are interested in not the characteristic of TCP itself, but the effect of avoidance of packet losses by DFC. It is worth noting that since the functions of DFC are independent of TCP, existence of TCP is not essential to the behavior of DFC itself.

The target flow and the background flow start at simulation time $t = 0$ s and $t = 0.1$ s, respectively. After the background flow traffic entered the network, the link from node 15 to 16 became a bottleneck, and traffic of both flows was regulated by predefined rules for DFC. For DFC, the values of transmission rate from external nodes in the other network to node 1 and node 15 is adjusted (shaped) to r_0^1 and r_0^2 as shown in (10). After congestion occurred, we investigated the temporal evolution of the network state.

In [7], we evaluated the performance of the DFC in a situation where there were many flows whose lifetime followed an exponential distribution. These flows were generated by two different nodes selected in the network model at random. In addition, we complemented DFC and TCP (TCP Tahoe, Reno, NewReno, Vegas) with respect to the number of hops in the other paper [9]. We confirmed that DFC achieved high performance and adapted to the network conditions.

5.2 Simulation Results

To demonstrate the diffusion effect by DFC and to verify the effect of the value of D, we show temporal evolution of the number of stored packet in each node.

Figure 7 shows the results obtained from our simulations with $D = 0.01$, 0.1, 0.4, 0.5, 1 and 3.0, respectively.

The horizontal axes denote node ID (1–29) and the vertical axes denote the number of stored packet at each node. Here, we omit node 30 since it has no stored packets. Simulation time is shown in each graph. Since the speed of diffusion effect strongly depends on the value of D, only two cases that $D = 0.4$ and 0.5, are displayed at the same timing.

After the time when the background traffic enters the simulation network (after 0.1 s), each node in the network prevents the stored packets centralizing at a certain node by DFC. From the results for $D = 0.01, 0.1, 0.4$ and 0.5, the distributions of the number of packets exhibit orderly behavior. Incidentally, if DFC was not applied, all stored packets are at node 15 [17]. This causes packet losses and the reduction of TCP window size. By introducing DFC, each node cooperatively acts to avoid packet losses even though decision-making of each node is based only on the local information. On the other hand, the results for $D = 1$ and 3 do not show orderly behavior of the packet distribution.

In general, the speed of diffusion effect increases when the value of D increases, but extremely large value of D prevents the diffusion effect. In these cases, although the value of $D = 0.5$ is out of the range of (27), the diffusion effects are observed.

Panels in Fig. 8 show the temporal evolution of the total number of packets, when the values of D are equal to $0.01, 0.1, 0.4, 0.5, 1$ and 3, respectively. The horizontal axes denote simulation time and the vertical axes denote the total number of packets that are stored in nodes. These results also indicates the speed of diffusion effect increases when the value of D increases, but extremely large value of D prevents the diffusion effect.

Next, we investigate how much packets are transmitted in the network. The volume of the target flow packet transmission at time t can be represented by the total number of the target flow packets in transit on links of the network. This quantity can represent amount of network performance at the present time, dedicated for the target flow. Panels in Fig. 9 show the results in cases $D = 0.01, 0.1, 0.4, 0.5, 1$ and 3, respectively. The horizontal axes denote simulation time and the vertical axes denote the total number of packets that are in transit on links.

Since the maximum number of packets in transit on a link at a moment was an average of 100, and the background flow passed through about half of links of the target flow, the maximum total number of target flow's packets in transit on links was about $2,900$ when $t \leq 0.1$ and was about $1,450$ when $t > 0.1$. On the other hand, the maximum total number of background flow's packets in transit on links was about 750 after $t = 0.1$.

From the first four panels for $D = 0.01, 0.1, 0.4$, and 0.5 in Fig. 9, the numbers of packets in transit on links for both flows reached almost their maximums in a short time and these results mean that they fairly share the link bandwidth. In the cases of $D = 1$ and 3 out of the range of (27), oscillation effects are observed. The reason of the oscillations is a trade-off between fast recovery and stability. For $D > 0.5$, it prefers too fast change of network state and prevents stable network performance as a result.

In these results, the value of $D = 0.5$ shows good performance but this is out of the range (27). The range (27) is derived from the sufficient condition that the diffusion effect appears. Thus, the value of $D = 0.5$ may exhibit good performance in some cases. However, in order to guarantee the diffusion effect for any conditions, it is necessary to adopt the value of D in the range (27). For example, we can choose $D = 0.499$.

We can say the range (27) is not a necessary condition but a sufficient condition for the guarantee of the diffusion effect in any network environment.

6. Conclusions

To overcome the difficulty in control of high-speed networks, we have proposed DFC. In this control mechanism, the state of the whole network is controlled indirectly through the autonomous action of each node; each node manages its local traffic flow on the basis of only the local information directly available to it, by using predetermined rules. By applying DFC, the distribution of the total number of packets in each node in the network becomes uniform over time, and it exhibits orderly behavior. This property is suitable for fast recovery from congestion.

One of important issues in design of DFC is how to choose the value of diffusion parameter. This is the central issue for enabling DFC to make diffusion effects of packet density in networks.

We determined the range of the diffusion parameter D by applying the condition for discrete space-time computations of the diffusion equation to DFC. In addition, we considered the actual value of D by dimensional analysis.

Simulation results verified the range of D. Even if the value is in the range, too small value of D causes very slow diffusion, and this means that stolid congestion recovery makes to waste much time. Consequently, to make fast diffusion, we should take a value of D as large as possible in this range.

Acknowledgement

This research was partially supported by the National Institute of Information and Communications Technology (NiCT) and the Grant-in-Aid for Scientific Research (S) No. 18100001 (2006–2010) from the Japan Society for the Promotion of Science.

References

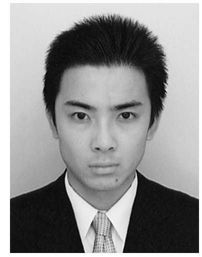
[1] Y. Bartal, J. Byers, and D. Raz, "Global optimization using local information with applications to flow control," Proc. 38th Ann. IEEE Symp. on Foundations of Computer Science, pp.303–312, Oct. 1997.

[2] S.H. Low and D.E. Lapsley, "Optimization flow control-I: Basic algorithm and convergence," IEEE/ACM Trans. Netw., vol.7, no.6, pp.861–874, 1999.

[3] K. Kar, S. Sarkar, and L. Tassiulas, "A simple rate control algorithm

for maximizing total user utility," Proc. IEEE INFOCOM 2001, pp.133–141, 2001.
[4] S. Kunniyur and R. Srikant, "A decentralized adaptive ECN marking algorithm," Proc. IEEE GLOBECOM'00, pp.1719–1723, 2000.
[5] J. Mo and J. Walrand, "Fair end-to-end window based congestion control," IEEE/ACM Trans. Netw., vol.8, no.5, pp.556–567, Oct. 1999.
[6] C. Takano and M. Aida, "Stability and adaptability of autonomous decentralized flow control in high-speed networks," IEICE Trans. Commun., vol.E86-B, no.10, pp.2882–2890, Oct. 2003.
[7] C. Takano, M. Aida, and S. Kuribayashi, "Autonomous decentralized flow control in high-speed networks with inhomogeneous configurations," IEICE Trans. Commun., vol.E87-B, no.6, pp.1551–1560, June 2004.
[8] C. Takano and M. Aida, "Diffusion-type autonomous decentralized flow control for end-to-end flow in high-speed networks," IEICE Trans. Commun., vol.E88-B, no.4, pp.1559–1567, April 2005.
[9] C. Takano, K. Muranaka, K. Sugiyama, and M. Aida, "Mutual complementarity of diffusion-type flow control and TCP," IEICE Trans. Commun., vol. E89-B, no.10, pp.2850–2859, Oct. 2006.
[10] J.B. Postel, "Transmission control protocol," Request for Comments 793, Sept. 1981.
[11] S. Floyd, "HighSpeed TCP for large congestion windows," Request for Comments 3649, Dec. 2003.
[12] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," ACM SIGCOMM Computer Communication Review, vol.32, April 2003.
[13] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control for fast long-distance networks," Proc. IEEE INFOCOM 2004, March 2004.
[14] C. Jin, D.X. Wei, and S.H. Low, "FAST TCP: Motivation, architecture, algorithms, performance," Proc. IEEE INFOCOM 2004, March 2004.
[15] I. Rhee and L. Xu, "CUBIC: A new TCP-friendly highspeed TCP variant," Proc. PFLDnet 2005, Feb. 2005.
[16] K.T. J. Song, Q. Zhang, and M. Sridharan, "Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks," Proc. PFLDnet 2006, Feb. 2006.
[17] C. Takano and M. Aida, "Diffusion-type autonomous decentralized flow control for multiple flows," IEICE Trans. Commun., vol.E90-B, no.1, pp.21–30, Jan. 2007.
[18] The Network Simulator — ns-2. http://www.isi.edu/nsnam/ns/

Chisa Takano received the B.E. degree in Telecommunications Engineering from Osaka University, Japan, in 2000, and received the Ph.D. in Information Network Engineering from Tokyo Metropolitan University, Japan, in 2008. In 2000 she joined the Traffic Engineering Center, NTT Advanced Technology Corporation (NTT-AT). She has been an Associate Professor of the Graduate School of Information Sciences, Hiroshima City University since April 2008. Her current interests include research and development of computer networks.



Keita Sugiyama received his M.E. degree in Management Systems Engineering from Tokyo Metropolitan University, Japan, in 2008.

Masaki Aida received his B.S. and M.S. degrees in Theoretical Physics from St. Paul's University, Tokyo, Japan, in 1987 and 1989, and received the Ph.D. in Telecommunications Engineering from the University of Tokyo, Japan, in 1999. In 1989, he joined NTT Laboratories. In April 2005 to March 2007, he was an Associate Professor of the Faculty of System Design, Tokyo Metropolitan University. He has been a Professor of the Graduate School of System Design, Tokyo Metropolitan University since April 2007. His current interests include traffic issues in computer communication networks. He is a member of the IEEE and the Operations Research Society of Japan.