PAPER

Diffusion-Type Autonomous Decentralized Flow Control for Multiple Flows*

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

SUMMARY We have proposed a diffusion-type flow control mechanism to achieve the extremely time-sensitive flow control required for high-speed networks. In this mechanism, each node in a network manages its local traffic flow only on the basis of the local information directly available to it, by using predetermined rules. In this way, the implementation of decision-making at each node can lead to optimal performance for the whole network. Our previous studies concentrated on the flow control for a single flow. In this paper, we propose a diffusion-type flow control mechanism for multiple flows. The proposed scheme enables a network to quickly recover from a state of congestion and to achieve fairness among flows.

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

1. Introduction

Due to the increasing popularity of broadband services, the construction of higher-speed backbone networks will be required in the near future. In a high-speed network, it is impossible to implement time-sensitive control based on collecting global information about the whole network because, even if the propagation delay is identical with that in low-speed networks, the state of each node varies rapidly with time, and is dependent on its processing speed. If we allow sufficient time to collect network-wide information, the data so gathered is too old to apply to 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:

- It must be possible to collect the information required for use in the control method.
- The control should take effect as rapidly as possible.

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

Decentralized flow control by end hosts, including TCP, is widely used in current networks, and there has been much research in this area [4]–[6]. However, since end-to-end or end-to-node control cannot be applied to decision-making in a time-scale shorter than the roundtrip delay, it is not capable of supporting decision-making on a very short time-scale. In low-speed networks, a control delay of the order of the round-trip time (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, although the RTT is itself unchanged, it becomes larger relative to the standard unit of time determined by the node's processing speed. This means that nodes in high-speed networks experience a larger RTT relative to the processing speed, and this causes an increase in the sensitivity to control delay. To achieve rapid control in a shorter time scale than the RTT, it is preferable for control to be applied by the nodes themselves rather than by the end hosts (see Fig. 1). Let us consider the situation that the RTT is 100 ms when a network is congested. The upper graph of Fig. 1 shows the relationship between the speed of the network and the number of packets that are influenced by the control delay, when flow control is applied by the end hosts. If the network speed is 10 Mbps, the number of packets influenced by the control delay from an end host is only several hundred. However, if the network speed is 100 Gbps, the number of packets is several million. Even though the RTT is unchanged, the increase in network speed has a severe impact on network performance. If we apply node-by-node control (the lower graph in Fig. 1), the control delay is reduced typically by a factor of 2000 compared to end host control.

We have therefore considered a control mechanism in which the nodes in a network handle their local traffic flows themselves, based only on the local information directly available to them. This mechanism can immediately detect a change in the network state around the node and implement quick decision-making. Although decision-making at a local node should lead to action suitable for managing the local performance of a network, it is not guaranteed that the

Manuscript received March 29, 2006.
Manuscript revised July 7, 2006.

[†]The author is with Traffic Engineering Division, NTT Advanced Technology Corporation (NTT-AT), Musashino-shi, 180-0006 Japan.

[††]The authors are with the Graduate School of System Design, Tokyo Metropolitan University, Hino-shi, 191-0065 Japan.

*An earlier version of this paper was presented at ITC19 [11].

a) E-mail: chisa.takano@ntt-at.co.jp

DOI: 10.1093/ietcom/e90–b.1.21

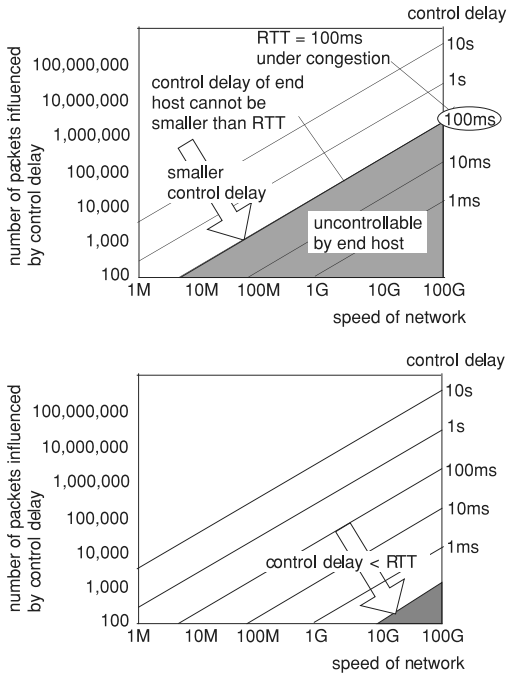

Fig. 1 Relationship between the number of packets influenced by the control delay and the speed of the network. The upper graph illustrates control by end hosts and the lower one illustrates node-by-node control (A trial calculation was made with an average distance between nodes of 10 km, average number of hops = 5, and a link utilization of 0.5).

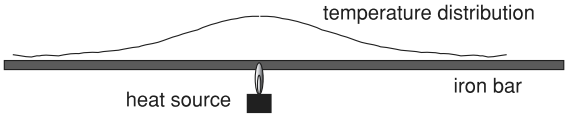

Fig. 2 Example of thermal diffusion phenomenon.

action is appropriate for the overall network-wide performance. So, the implementation of decision-making at each node may not lead to optimal performance for the whole network.

In our previous studies, we investigated the behavior of local packet flows and the global performance when a node is congested, and proposed a diffusion-type flow control (DFC) [7], [8]. In addition, we investigated the stability and adaptability of the network performance when the capacity of a link is changed, by using network models with homogeneous [8] and inhomogeneous [9] configurations. DFC provides a framework in which the implementation of decision-making at each node leads to high performance for the whole network. The principle of our flow control model can be explained through the following analogy [10].

When we heat a point on a cold iron bar, the temperature distribution forms a normal distribution and heat spreads through the whole as a diffusion phenomenon (Fig. 2). In this process, the action in a minute segment of the iron bar is very simple: heat flows from the hotter side towards the cooler side. The rate of heat flow is proportional to the temperature gradient. There is no direct 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 those in an adjacent node. Then the distribution of the number of packets in each 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. In our previous studies, we have focused on the flow control for a single flow and the packet density has been made uniform along a one-dimensional path of the flow. In this paper, we focus on the environment of multiple flows and we consider our goal to be the equalization of the packet density. We propose DFC for multiple flows, which can achieve the equalization of packet density along a one-dimensional path. The effective characteristics of the proposed control method are evaluated through simulations.

2. Preliminary Description of Flow Control for a Single Flow

In the case of Internet-based networks, to guarantee the end-to-end quality of service (QoS) of a flow, a QoS-sensitive flow uses a static route (e.g. RSVP). Thus, we assume that the target flow has a static route.

In addition, we assume that all routers in the network can employ per-flow queuing for all the target flows[†]. Our flow control for a single flow consists of two parts: the main flow control part and the optional traffic regulation part. The main part is DFC and this works to equalize the number of packets in routers along the path of a flow. The optional part regulates the volume of traffic at the ingress point of the network. This function is not mandatory and can be replaced with other types of flow control, e.g., the window control of TCP.

2.1 Diffusion-Type Flow Control

In DFC, each node controls its local packet flow autonomously, and as a result, the distribution of the total number of packets in nodes along the path of a flow becomes uniform over time.

Figure 3 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, ...) transfers packets to node $i+1$ and node $i+1$ sends feedback information to node i. For simplicity, we assume that packets have a fixed length in bits.

[†]The assumption of per-flow queuing is not mandatory in the framework of DFC, but it is convenient to use it to simplify the explanation of the framework. In actual fact, it is hard to implement per-flow queuing in high-speed networks. Fundamentally, DFC only requires "per-input port" queueing.

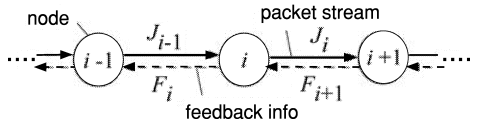

Fig. 3 Node interactions in our flow control model.

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

When node i receives feedback information from downstream node $i+1$, it determines the appropriate transmission rate for packets to the downstream node $i+1$ using the received feedback information, and adjusts its transmission rate towards the downstream node $i+1$ accordingly. The framework for node behavior and flow control may be summarized as follows:

- Each node i autonomously determines the transmission rate J_i only on the basis of 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.
  (If there are no packets 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 at a fixed interval τ_i for each link.
- 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 adjacent nodes. There is no centralized control mechanism in the network.

Next, we will explain the details of the DFC. The transmission rate $J_i(\alpha, t)$ of node i at time t is determined by

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

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

where L_i denotes the value of the link capacity from node i to node $i+1$, $n_i(t)$ denotes the number of packets in node i at time t, $r_i(t-d_i)$ is the target transmission rate specified by the downstream node $i+1$ as feedback information, and d_i denotes the propagation delay between node i and node $i+1$.

In addition, $r_i(t-d_i)$ and $n_{i+1}(t-d_i)$ are notified at regular intervals with a fixed period τ_{i+1} from the downstream node $i+1$ with a propagation delay d_i. Parameter α (≥ 1) is a constant and is the flow intensity multiplier. Parameter D_i is chosen to be inversely proportional to the propagation delay [9] as follows:

$$D_i = D\,\frac{1}{d_i}, \tag{3}$$

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

The feedback information $\mathbf{F}_i(t)$, created by node i at regular fixed intervals of period τ_i, consists of the two quantities shown in (4):

$$\mathbf{F}_i(t) = (r_{i-1}(t), n_i(t)). \tag{4}$$

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

$$r_{i-1}(t) = J_i(1, t). \tag{5}$$

Moreover, the packet flow $J_i(t)$ in node i is adjusted whenever feedback information arrives from the downstream node $i+1$ (with a periodicity of $\tau_{i+1} = d_i$).

To allow an intuitive understanding, we will briefly explain the physical meaning of DFC. Let us replace i with x and apply continuous approximation. Then the propagation delay becomes $d_i \to 0$ for all i and the packet flow (2) may be expressed as

$$\tilde{J}(\alpha, x, t) = \alpha\, r(x, t) - D\,\frac{\partial n(x, t)}{\partial x}, \tag{6}$$

and the temporal variation of the packet density $n(x, t)$ may be expressed as a diffusion-type equation,

$$\frac{\partial n(x, t)}{\partial t} = -\alpha\,\frac{\partial r(x, t)}{\partial x} + D\,\frac{\partial^2 n(x, t)}{\partial x^2}, \tag{7}$$

using the continuous equation

$$\frac{\partial n(x, t)}{\partial t} = -\frac{\partial \tilde{J}(\alpha, x, t)}{\partial x}. \tag{8}$$

That is, our method aims to perform flow control using the analogy of a diffusion phenomenon. We can therefore expect excess packets in a congested node to become distributed over the whole network and 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 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 assume that the external nodes only have a traffic shaping function, which can adjust the transmission rate to the requested rate notified from the downstream node. That is,

[†]The range of D is described in Appendix.

an external node 0 cannot calculate the transmission rate $J_0(\alpha, t)$ using Eq. (2), but can adjust its transmission rate to $r_0(t - d_0)$, which is notified from node 1.

We consider a rule for determining $r_0(t)$ as a boundary condition. Node 1 can calculate $J_0(\alpha, t)$ if we assume the number of packets stored in the other networks' node is 0. The target rate $r_0(t)$, notified from node 1, is created as $\tilde{J}_0(\alpha, t)$ with the above assumption. That is,

$$r_0(t) := \tilde{J}_0(\alpha, t + d_0) = \alpha J_1(1, t) - D_0 n_1(t). \quad (9)$$

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

2.2 Packet-Rate Regularization at the Ingress to Networks

Although DFC excels in quick equalization of packet density, there is nothing to prevent an excessive packet flow coming from outside the network, because DFC is based only on autonomous operation of nodes using local information. So, we need to combine a packet shaping function with DFC, if necessary.

The packet shaping is not a local control but is based on global network information, that is, the ingress node regulates the transmission rate to be less than or equal to the minimum value of the available link capacity of all the downstream links [10].

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 adjusts its transmission rate towards the downstream node $i + 1$.

Node i's packet transmission rate to the downstream node $i + 1$ is determined as (2). In addition, node i generates feedback information $\mathbf{F}_i(t)$ as

$$\mathbf{F}_i(t) = (r_{i-1}(t), n_i(t), \ell_i(t)), \quad (10)$$

and notifies this information to the upstream node $i - 1$.

The feedback information is expressed as:

$$r_{i-1}(t) = J_i(1, t), \quad \text{and} \quad (11)$$

$$\ell_i(t) = \min(L_i, \ell_{i+1}(t - d_i)). \quad (12)$$

The newly added information $\ell_i(t)$ is not used for the determination of transmission rate $J_i(\alpha, t)$ at the network nodes ($i = 1, 2, \ldots, N$), but it is used only for the regularization of the transmission rate at the ingress point of the network. The packet shaping rate at the ingress, $J_0(\alpha, t)$, is determined as

$$\begin{aligned} J_0(\alpha, t) &= \max(0, \min(\ell_0(t), r_0(t - d_0))), \\ &= \max(0, \min(\ell_0(t), \tilde{J}_0(\alpha, t))). \end{aligned} \quad (13)$$

Note that $\ell_0(t)$ is calculated from $\ell_1(t-d_0)$ notified from node 1 and from the bandwidth of access link L_0 (its propagation delay being d_0). So, it is not necessary to calculate Eq. (13) at the node outside of the network. If the node outside the network does not support DFC, it cannot calculate Eq. (13). In this case, node 1 notifies $\min(\ell_1(t), r_0(t))$ as a shaping rate to the node outside of the network. Then, the node outside the network should be capable of regulating the input traffic according to $\min(\ell_1(t), r_0(t))$ notified from node 1.

3. Classification of Types of Equalization of Distribution

In this section, we consider the equalization of the packet density and classify it into types.

The equalization of packet density may be classified into the following types.

(a) Serial Diffusion
To avoid packet loss, the number of packets of the target flow in routers along the path of the target flow is equalized. This type may be further divided into two types.

(a-1) Backward Serial Diffusion
With respect to a bottleneck link, the number of packets is equalized towards the upstream direction.

(a-2) Forward Serial Diffusion
With respect to a bottleneck link, the number of packets is equalized towards the downstream direction.

(b) Parallel Diffusion
For multiple flows which share all or a part of the path, the number of packets at a node on the common path is made to be equally distributed among all the flows.

In our previous work, our target was the realization of serial diffusion, in particular, backward serial diffusion. This is because, for a single flow environment, a bottleneck link with a bandwidth L_i prevents the packet flow at node i from being greater than L_i (see Sect. 5.2). However, in a multiple flow environment, the bandwidth of the bottleneck link is shared by multiple flows. So, it is possible that some flows have a larger rate than others. For short-time congestion recovery, both backward and forward serial diffusion are important. We can expect that both may be achieved by appropriate setting of the available bandwidth of each flow.

We hope that the effect of packet equalization will spread not only along the path of the target flows but also to the whole network, and for this purpose the equalization of packets between flows by parallel diffusion is important. To realize parallel diffusion, we assign an appropriate available bandwidth L_i for each flow passing through node i.

From the above considerations, the appropriate setting of L_i is the key issue in achieving DFC with backward, forward and parallel diffusion effects.

4. Diffusion-Type Flow Control for Multiple Flows

4.1 Framework

In this paper, all flows are in the same priority class and it is

desirable that all active flows share the link bandwidth fairly. Extension to the case where each flow requires a different bandwidth is easy.

Assume that there are M_i flows sharing the link between node i and $i+1$, and they are identified by j ($j = 1, 2, \ldots, M_i$). Some quantities are redefined for each flow j as follows.

$n_i^j(t)$: the number of packets belonging to flow j in node i at time t.

$r_i^j(t-d_i)$: the notified rate for flow j by using feedback information from the downstream node $i+1$. It is notified with propagation delay d_i.

$n_{i+1}^j(t-d_i)$: the notified number of packets belonging to flow j in node $i+1$ obtained from feedback information from the downstream node $i+1$. It is notified with propagation delay d_i.

$L_i^j(t)$: the available bandwidth for flow j of the link from node i to node $i+1$.

Using these quantities, we consider the framework of DFC for multiple flows. When the feedback information from downstream node $i+1$ is received, each node i autonomously determines the transmission rate $J_i^j(\alpha,t)$ on the basis only of the local information directly available to it. In addition, each node i autonomously creates feedback information $\mathbf{F}_i^j(t)$ according to a predefined rule and sends it to the upstream node $i-1$. The interval for generating feedback information is proportional to the propagation delay, d_{i-1}, between nodes $i-1$ and i. Let the average number of active flows in node i observed between the last two successive occurrences of feedback information generation be $m_i(t)$. That is, $m_i(t)$ is the average number of distinct flows of packets in node i. In general, $m_i(t) \le M_i$. Let the link bandwidth from node i to node $i+1$ be B_i. Then, the transmission rate for flow j, $J_i^j(\alpha,t)$, is determined as

$$J_i^j(\alpha,t) = \max(0, \min(L_i^j(t), \tilde{J}_i^j(\alpha,t))), \tag{14}$$

$$\tilde{J}_i^j(\alpha,t) = \alpha\, r_i^j(t-d_i) - D_i\,(n_{i+1}^j(t-d_i) - n_i^j(t)). \tag{15}$$

Feedback information for flow j generated by node i consists of

$$\mathbf{F}_i^j(t) = (r_{i-1}^j(t), n_i^j(t)), \tag{16}$$

and is notified to the upstream node $i-1$. Here, the target rate for flow j is

$$r_{i-1}^j(t) = \max(0, \min(B_i/m_i(t), \tilde{J}_i^j(1,t))). \tag{17}$$

When node i is at the ingress of the network ($i = 1$), node 1 notifies the following information $r_0^j(t)$ to the upstream node or end host (outside of the network).

$$r_0^j(t) := \alpha\, J_1^j(1,t) - D_0\, n_1^j(t)\ (= \tilde{J}_0^j(\alpha, t+d_0)) \tag{18}$$

If packet shaping at the ingress to the network is required, we need to add the following rules to the above framework. We add the information of the available bandwidth for flow j to the feedback information for flow j generated at node i,

$$\mathbf{F}_i^j(t) = (r_{i-1}^j(t), n_i^j(t), \ell_i^j(t)), \tag{19}$$

and notify this to the upstream node $i-1$. Information of the available bandwidth $\ell_i^j(t)$ is generated as

$$\ell_i^j(t) = \min(B_i/m_i(t), \ell_{i+1}^j(t-d_i)). \tag{20}$$

$\ell_i^j(t)$ is used only for packet shaping at the ingress to the network, and the packet shaping rate $J_0^j(\alpha,t)$ is determined as

$$\begin{aligned} J_0^j(\alpha,t) &= \max(0, \min(\ell_0^j(t), r_0^j(t-d_0))), \\ &= \max(0, \min(\ell_0^j(t), \tilde{J}_0^j(\alpha,t))). \end{aligned} \tag{21}$$

4.2 Determination of Available Bandwidth

To achieve bidirectional serial diffusion and parallel diffusion in the framework of DFC for multiple flows, we adjust the available bandwidth L_i^j appropriately between flows.

Let the bandwidth of the link from node i to node $i+1$ be B_i. If we choose $L_i^j(t)$ ($j = 1, 2, \ldots, M_i$) as large as possible, they must satisfy

$$\sum_{j=1}^{m_i(t)} L_i^j(t) = B_i. \tag{22}$$

If we choose $L_i^j(t)$ to have the fixed value $B_i/m_i(t)$, interference between flows does not occur, and both the forward serial diffusion and parallel diffusion are not realized. This is because the flow control for each flow is reduced to that for a single flow.

In DFC, the ideal transmission rate is $\tilde{J}_i^j(\alpha,t)$, and $J_i^j(\alpha,t)$ is restricted by the available bandwidth. In particular, if there are a lot of flows, the ideal transmission rate $\tilde{J}_i^j(\alpha,t)$ is frequently governed by the inequality

$$\sum_{j=1}^{m_i(t)} \tilde{J}_i^j(\alpha,t) > B_i, \tag{23}$$

and the probability of each flow getting its ideal transmission rate $\tilde{J}_i^j(\alpha,t)$ is low. This prevents the smooth equalization of the packet density. So, we take the following two conditions

- $L_i^j(t)$ is chosen as proportional to $\tilde{J}_i^j(\alpha,t)$ for each j, and
- $\sum_{j=1}^{m_i(t)} L_i^j(t) = B_i$ (24)

into consideration.

Then, the simplest way to determine L_i^i is

$$L_i^j = B_i\, \frac{\tilde{J}_i^j(\alpha,t)}{\sum_{j=1}^{m_i(t)} \tilde{J}_i^j(\alpha,t)}. \tag{25}$$

That is, the bandwidth B_i is shared by flow with a weight $\tilde{J}_i^j(\alpha,t)$.

This rule means that flows with larger $\tilde{J}_i^j(\alpha, t)$ can obtain a larger relative transmission rate, and so can transmit a relatively larger volume of traffic to the downstream node. Then, the transmission rates of other flows are regulated to smaller values. We can expect this to allow the implementation of both forward serial diffusion and parallel diffusion.

5. Simulation Results

This section demonstrates the characteristics of DFC by giving the results of simulations. First, we show comparisons between the simulation results of DFC for multiple flows and that for a single flow. Next, we show comparisons between DFC and TCP. In our evaluation, we adopt packet shaping using $\ell_i(t)$, the parameters of DFC are set as $D = 0.1$ and $\mathbf{F}_i^j(t)$, and the interval of the feedback information $\mathbf{F}_i^j(t)$ is d_{i-1}.

5.1 Evaluation Model

Figure 4 shows our network model with 60 nodes, which was used in the simulations. Although each 1-dimensional model looks simple, it represents a part of a network and describes a path of a target end-to-end flow extracted from the whole network. We represent the lengths of links by their delays, and choose this to be 0.1 ms, which is taken as the unit of time. The packets have a fixed length and the link bandwidth is 100 packets per unit of time.

To allow direct comparison with our previous results, the network model is the same as that used in our previous research [10].

The simulation scenario is as follows. There are two flows. One is the target flow and the other is a background flow. The target flow begins at time $t = 0.1$ s and the background flow begins at time $t = 0$ s. The path of the target flow is from node 1 to node 60 and that of the background flow is from node 30 to node 60. The maximum rate (without traffic regulation) of both flows is 100 packets per unit time (the same as the link bandwidth).

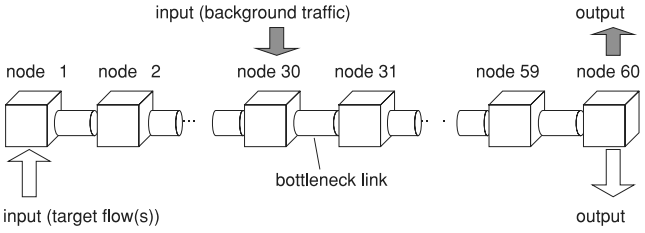

Fig. 4 Network model.

Both flows have greedy traffic, that is, the rate of each flow is as large as possible. If we use a bursty traffic model, the volume of the input traffic is less than in our greedy model. After the target flow traffic starts entering the network, the link from node 30 to node 31 becomes a bottleneck, and the traffic of both flows is regulated by the predefined rules for DFC and packet shaping. After congestion occurs, we investigate the temporal evolution of the network state, for both DFC for multiple flows and DFC for a single flow.

5.2 Simulation Results for Serial Diffusion

Figure 5 shows the temporal evolution of the total number of packets stored in each node, for the target flow, when DFC for a single flow is applied. We choose the available bandwidth at the bottleneck link as

$$L_{30}^1 = L_{30}^2 = B_{30}/2, \tag{26}$$

where flows 1 and 2 represent the target and background flows, respectively. This means the target and background flows share the bandwidth of $B_{30} = 100$ packets per unit of time equally.

Because the transmission rate $J_{30}^1(1, t)$ is regulated to be less than or equal to $B_{30}/2$ at node 30, the nodes downstream of node 30 are not congested. Through the effect of backward serial diffusion, the congestion at node 30 is spread to the upstream nodes and prevents packet loss at node 30. If we do not apply DFC, all stored packets are concentrated at node 30 and this might cause packet loss.

Figure 6 shows the result in the case where DFC for multiple flows is applied. The available bandwidth at the bottleneck link is determined by (25), and its value changes dynamically depending on the situation. From this result, we can see both backward and forward serial diffusion. We can also see that the time to recover from congestion is shorter than that of the case for a single flow.

The occurrence of the forward serial diffusion is caused by the effect that the bandwidth available to the background flow 2 is set to $L_{30}^2 < B_{30}/2$. We need also to consider the behaviors of flow 2: the investigations of both the number of stored packets and the volume of packet transmission.

An evaluation of the first of these is described in the next subsection with result of parallel diffusion, while results for the second of these are described here. The packet transmission volume at time t can be denoted by the number

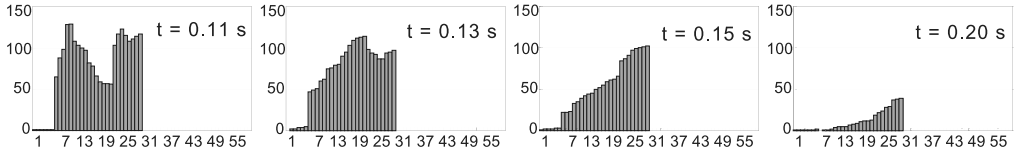

Fig. 5 Temporal evolution of the number of packets stored in nodes controlled by DFC for a single flow. The horizontal axis denotes node ID and the vertical axis denotes the number of stored packets.

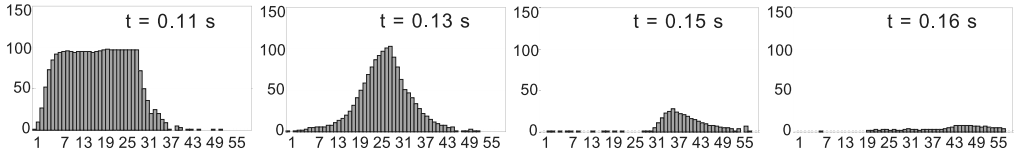

Fig. 6 Temporal evolution of the number of packets stored in nodes controlled by DFC for multiple flows. The horizontal axis denotes node ID and the vertical axis denotes the number of stored packets.

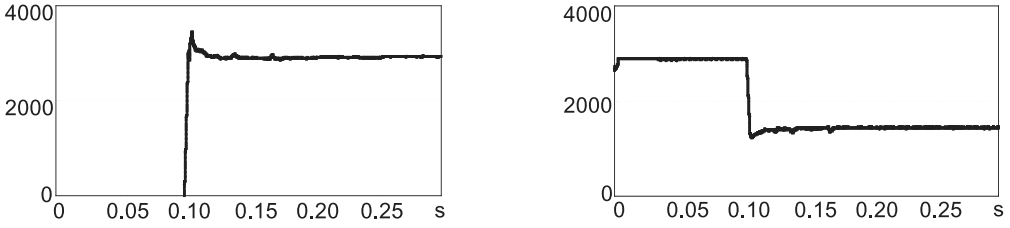

Fig. 7 Temporal evolutions of the total number of packets in transit on links. The horizontal axis denotes simulation time and the vertical axis denotes the number of packets in transit.

of packets in transit on the links of the network. Figure 7 shows the results of the simulation. Since the maximum number of packets in transit on a link at any one time is 100, the maximum total number of flow 1 packets in transit on links is 3000 after $t = 0.1$ s. In the case of flow 2, since flow 2 passes through about half of the links of flow 1, the maximum total number of flow 2 packets in transit on links is 3000 when $t \leq 0.1$ s and is 1500 when $t > 0.1$ s. From Fig. 7, the numbers of packets in transit on links for both flows reach almost their maximum in a short time and these results mean they share the link bandwidth fairly.

5.3 Evaluation of Parallel Diffusion

This subsection shows the temporal evolution of the number of packets belonging to each flow stored in the node at which all the flows come together. The network model used in this evaluation is the same as in Sect. 5.1.

The simulation scenario is as follows. There are 32 flows. Half of these, flows 1–16, are target flows, and the path of the target flows is from node 1 to node 60. The target flows begin at time time $t = 0$ s. The other flows, flows 17–32, are background flows and the path of the background flows is from node 30 to node 60. The background flows start at different times. The first flow starts at $t = 0.1$ s and the interval between the start time of subsequent flows is 0.1 s. That is, flow 17 starts at time $t = 0.1$ s, flow 18 starts at time $t = 0.2$ s, and flow 32 starts at time $t = 1.6$ s. Although the maximum rate (without traffic regulation) of each flow is 100 packets per unit of time (the same as link bandwidth), the actual traffic of each flow is regulated by the predefined rules of DFC and packet shaping[†]. After the traffic of the background flow 17 starts entering the network, the link from node 30 to node 31 becomes a bottleneck.

Figure 8 shows the temporal evolution of the number of packets of each flow stored in the congested node 30. The horizontal axis denotes the flow ID and the vertical axis denotes the number of stored packets. In particular, Fig. 8 illustrates the position immediately after flows 17, 18, 19, and 32 start.

Immediately after each background flow starts, we can see that the number of stored packets for the new flow becomes large. Simultaneously, the total number of packets for the existing flows also becomes large. After that, the number of packets for the new flow decreases and becomes uniform with that of the other existing flows.

Figure 8 implies that DFC for multiple flows enables rapid recovery from congestion by influencing the state of existing flows. However, since the numbers of packets belonging to existing flows which are stored in node 30 are sufficiently small, the influence on existing flows is small.

Now, as in the previous section, we investigate the number of packets in transit on the links of the network, in order to investigate the efficiency of the network. Figure 9 shows results for three of the target flows and three of the background flows as typical examples. Since 32 flows share the network link, the maximum number of packets in transit on a link is about 190 for target flows, and is 95 for background flows, when $t > 1.6$ s.

From Fig. 9, it can be seen that the number of packets in transit on links for both types of flow are reduced to be close to their maximum levels in a short time and these results indicate that they fairly share the link bandwidth.

5.4 Comparisons between DFC and TCP

In order show the characteristics of DFC in a realistic situation, we compare DFC and TCP. We use a network model similar to that shown in Fig. 4 but smaller, with only 16 nodes. The buffer capacity at each node is 1800 packets.

[†]Incidentally, although each flow has greedy traffic, the aggregation of the background flows can be regarded as some bursty traffic when we consider one of the target flows.

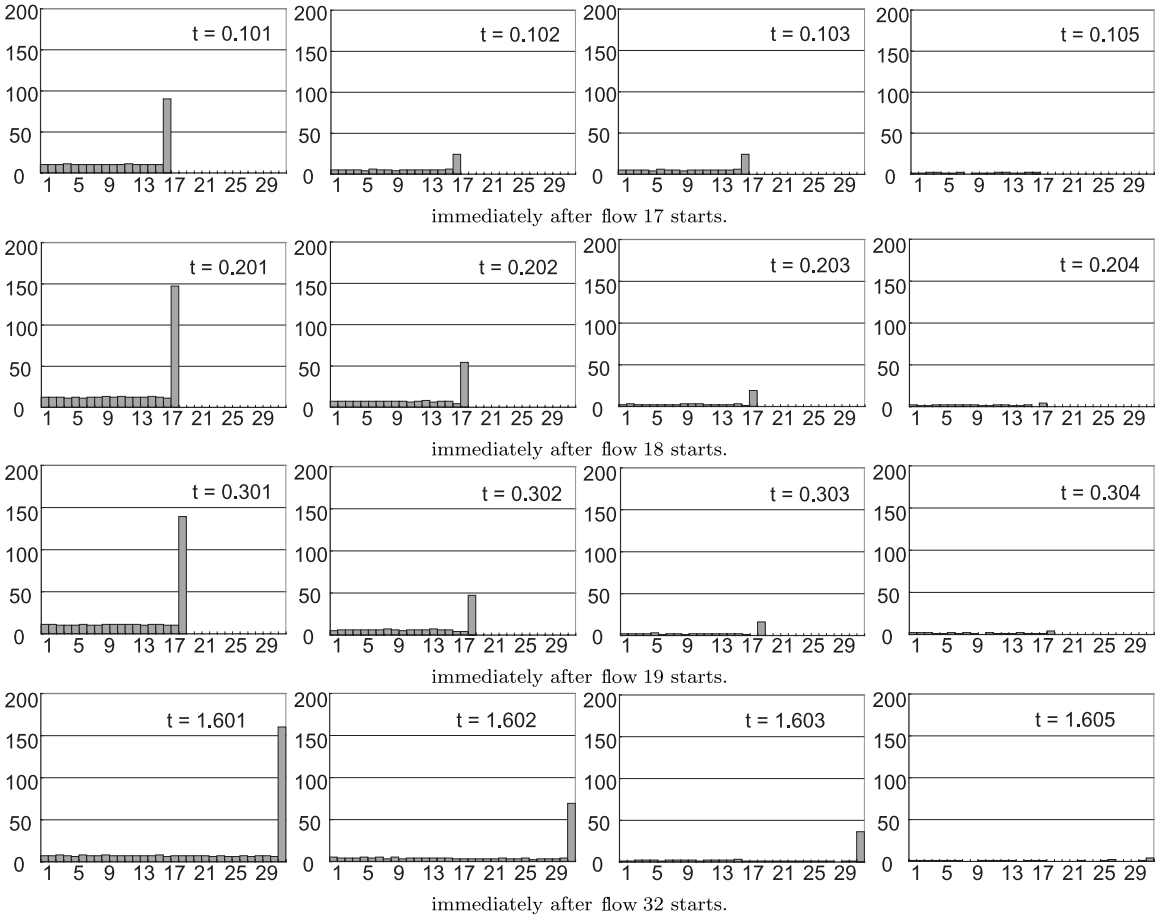

Fig. 8 The number of packets stored in node 30 for each of the 30 flows at various times.

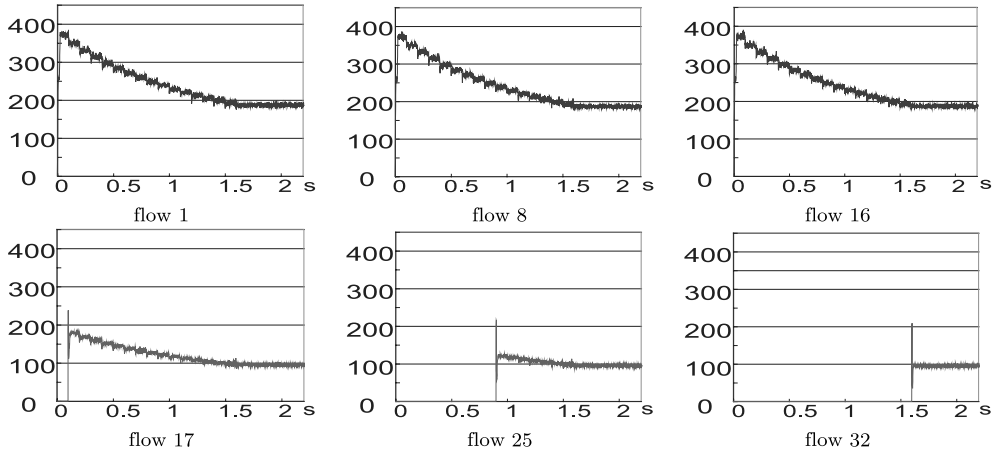

Fig. 9 Variation with time of number of packets stored in node 30 for six of the 32 flows.

The target flow is between node 1 and node 16, while the background traffic flows between node 8 and node 16. The target flow and the background flow start at simulation times $t = 0$ s and $t = 0.1$ s, respectively. Both flows are controlled by TCP Reno or DFC. The maximum window size of TCP is sufficiently large with respect to the bandwidth-delay product of the end-to-end routes.

First, to investigate the difference between TCP and DFC, we show the temporal evolution of the number of packets stored in each node. Figures 10 and 11 show the results obtained for TCP and DFC, respectively. The horizontal axes denote node ID (1–16) and the vertical axes denote the number of packets stored at the node. The five different graphs for each case represent different instants during the

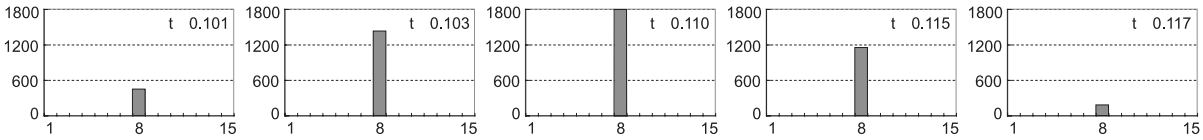

Fig. 10 Temporal evolution of distribution of packets stored in each node (TCP Reno).

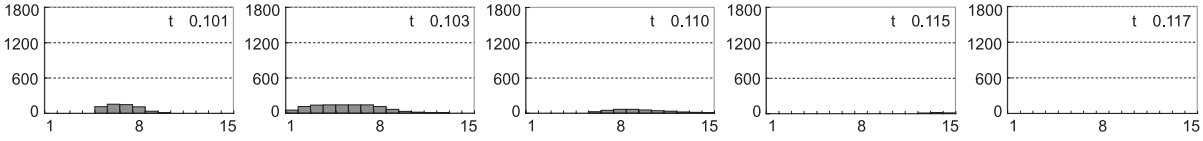

Fig. 11 Temporal evolution of distribution of packets stored in each node (DFC for multiple flows).

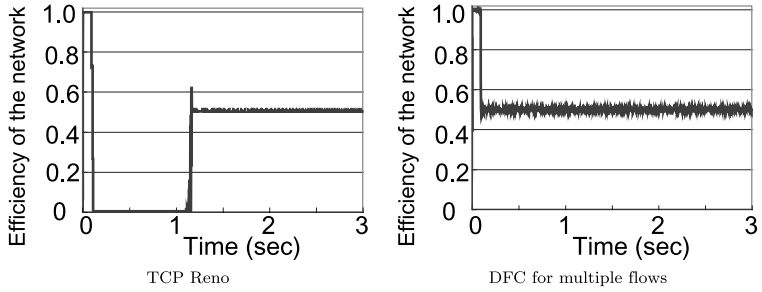

Fig. 12 Efficiency of the networks.

simulation, the time being shown on each graph.

In the case of Fig. 10, after the instant when the background traffic started ($t = 0.1$ s), all the stored packets were at node 8, leading to packet loss. The number of stored packets at node 8 reached the buffer size of 1800 and packet loss then occurred. After that, the TCP window size was reduced and the number of stored packets decreased. In the case of Fig. 11, DFC prevented the stored packets from building up at a particular node. Due to the operation of DFC, packet loss was avoided. Through the introduction of DFC, each node acts cooperatively to avoid packet loss even though the decision-making of each node is based only on the local information.

Figures 12 shows the efficiency of the networks. The horizontal axes denote the simulation time and the vertical axes denote the efficiency of the networks. Here, the efficiency of the network is represented by the normalized value of the total number of packets that are in transit on links, that is, the ratio of the total number of packets that are in transit on links to the maximum number of packets that could be in transit on links. The total number of packets means the number of packets being transported by the network at a particular instant. The left-hand graph shows the result obtained using TCP Reno, and the right-hand graph shows the result obtained using DFC for multiple flows. After the time when the background traffic started (after 0.1 s), the available bandwidth for the target flow was reduced to a half. Since the efficiency of the network reaches 0.5 within 0.1 s, the target flow is sharing the link bandwidth fairly, with high utilization.

The end-to-end packet delay in the network is affected by the number of packets stored at nodes. From Fig. 11, the number of the stored packets can be seen to decrease rapidly with DFC. This means the waiting time element of the end-to-end delay approaches zero quickly. In addition, Fig. 12 shows that DFC achieves almost ideal throughput. Thus, with the use of DFC and traffic shaping, the end-to-end packet delay is minimized.

6. Conclusions

In this paper, we have considered the extension of DFC so that it supports multiple flows.

DFC aims to prevent the concentration of packets at a congested node and so to avoid packet loss. The means of equalizing the number of packets may be classified into two main types. One is serial diffusion along the path of a flow, and the other is the parallel diffusion between different flows. Our previous work was concerned with DFC for a single flow and it considered only one aspect of serial diffusion (backward serial diffusion). By choosing the available bandwidth for each flow appropriately, DFC for multiple flows can achieve not only backward serial diffusion, but also forward serial diffusion and parallel diffusion. In addition, the proposed flow control can reduce the time to recover from congestion.

Acknowledgments

The authors would like to thank Keita Sugiyama, a grad-

uate student at Tokyo Metropolitan University, for help in performing some of the simulation experiments. This research was partially supported by 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] 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.
[5] S. Kunniyur and R. Srikant, "A decentralized adaptive ECN marking algorithm," Proc. IEEE GLOBECOM'00, pp.1719–1723, 2000.
[6] R. Johari and D. Tan, "End-to-end congestion control for the Internet: Delays and stability," IEEE/ACM Trans. Netw., vol.9, no.6, pp.818–832, Dec. 2001.
[7] M. Aida and C. Takano, "Stability of autonomous decentralized flow control schemes in high-speed networks," Proc. IEEE ICDCS 2002 Workshop (ADSN 2002), pp.63–68, July 2002.
[8] 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.
[9] 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.
[10] 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.
[11] M. Aida, C. Takano, and A. Miura, "Diffusion-type flow control scheme for multiple flows," The 19th International Teletraffic Congress (ITC19), pp.133–142, 2005.

Appendix: Range of Diffusion Parameter D

The partial differential Eq. (7) describes the temporal evolution of packet density in the context of continuous approximation of networks. The first term on the right-hand side in (7) describes a sustainable packet flow and this is not concerned with diffusion. The parameter setting of $\alpha = 1$ is appropriate for balancing between input and output traffic at a node [10].

On the other hand, the second term on the right-hand side in (7) is essential in the diffusion effect. Thus, we omit the first term and consider the following partial differential equation,

$$\frac{\partial n(x,t)}{\partial t} = D\,\frac{\partial^2 n(x,t)}{\partial x^2}, \qquad (\mathrm{A}\cdot 1)$$

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

For simplicity, we assume all the links in networks have the same length $\Delta x = 1$. In this situation, the interval of the DFC action is the same for all nodes, and we denote it as $\Delta t = 1$. The difference equation corresponding to (A· 1) is as follows:

$$n(x,t+1) - n(x,t) = D\,\{n(x+1,\,t) - 2\,n(x,\,t) + n(x-1,\,t)\}\,. \qquad (\mathrm{A}\cdot 2)$$

If the solution of (A· 2) exhibits similar behavior to that of (A· 1), DFC works appropriately and causes the diffusion of packet density. Our concern is to find the appropriate value of D in which the solution of (A· 2) exhibits the diffusion phenomenon. We can express $n(x, t+1)$ as

$$n(x,t+1) = D\,n(x+1,\,t) + (1-2D)\,n(x,\,t) + D\,n(x-1,\,t). \qquad (\mathrm{A}\cdot 3)$$

If $D < 1/2$, (A· 3) means the temporal evolution of $n(x,t)$ is obtained from the weighted average of $n(x,t)$ around x. This constraint of D is the same as the constraint that appears in solving (A· 1) by discrete space-time computation. Thus, the range of D should be $0 < D < 1/2$. In our evaluation, we chose the value of D in this range and DFC exhibited the diffusion effect of the packet density.

Chisa Takano received the B.E. in Telecommunications Engineering from Osaka University, Japan, in 2000. In 2000 she joined Traffic Research Center, NTT Advanced Technology Corporation (NTT-AT). She has been engaged in research and development of computer networks. She received the Young Researcher's Award of IEICE in 2003.

Masaki Aida received his B.S. and M.S. in Theoretical Physics from St. Paul's University, Tokyo, Japan, in 1987 and 1989, respectively, and received the Ph.D. in Telecommunications Engineering from the University of Tokyo, Japan, in 1999. Since joining NTT Laboratories in 1989, he has been mainly engaged in research on traffic issues in computer communication networks. From 1998 to 2001, he was a manager at Traffic Research Center, NTT Advanced Technology Corporation (NTT-AT). Since April 2005, he has been an Associate Professor at the Faculty of System Design, Tokyo Metropolitan University. His current interests include traffic issues in communication systems. Dr. Aida received the Young Researcher's Award of IEICE in 1996. He is a member of the IEEE and the Operations Research Society of Japan.