PAPER

Diffusion-Type Autonomous Decentralized Flow Control for End-to-End Flow in High-Speed Networks

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

SUMMARY We have proposed diffusion-type flow control as a solution for the extremely time-sensitive flow control required for high-speed networks. In our method of flow control, we design in advance simple and appropriate rules for action at the nodes, and these automatically result in stable and efficient network-wide performance through local interactions between nodes. Specifically, we design the rules for the flow control action of each node that simulates the local interaction of a diffusion phenomenon, in order that the packet density is diffused throughout the network as soon as possible. However, in order to make a comparison with other flow control methods under the same conditions, the evaluations in our previous studies used a closed network model, in which the number of packets was unchanged. This paper investigates the performance of our flow control method for an end-to-end flow, in order to show that it is still effective in more realistic networks. We identify the key issues associated with our flow control method when applied to an open network model, and demonstrate a two-step solution. First, we consider the rule for flow control action at the boundary node, which is the ingress node in the network, and propose a rule to achieve smooth diffusion of the packet density. Secondly, we introduce a shaping mechanism, which keeps the number of packets in the network at an appropriate level.

key words: *flow control, diffusion, autonomous decentralized control, high speed network*

1. Introduction

We have previously proposed diffusion-type flow control (DFC) as a solution for the extremely time-sensitive flow control required for high-speed networks [1], [2]. The motivation of our work on flow control is to introduce a new framework of time-sensitive network control for high-speed networks.

Delay in communication networks consists of processing delay and propagation delay. Processing delay is the delay experienced by a packet while waiting at nodes for transmission. This may be reduced if the processing speed of the nodes is increased. Propagation delay is the packet delay experienced during propagation on links and is determined by the link length and the speed of light. Unlike processing delay, the propagation delay is fixed even if the processing speed of the nodes is increased. So, if the processing speed of the nodes is made high enough, propagation delay becomes the dominant part of the total delay.

Manuscript received May 12, 2004.

Manuscript revised October 29, 2004.

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

††The author is with the Faculty of System Design, Tokyo Metropolitan University, Hino-shi, 191-0065 Japan.

a) E-mail: chisa@m.ieice.org

b) E-mail: aida@m.ieice.org

DOI: 10.1093/ietcom/e88–b.4.1559

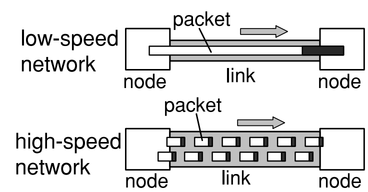

Fig. 1 Effect of large bandwidth-delay product.

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, depending on its processing speed, while the propagation delay is constant. If we allow sufficient time to collect network-wide information, the gathered data is too old to apply to time-sensitive control. In this sense, each node in a high-speed network is isolated from information about the state of other nodes or of the overall network [3], [4]. In addition, when the propagation delay is dominant, at any instant a large amount of data exists on the links. Figure 1 shows situations where packets are transmitted in low-speed and in high-speed networks. In a low-speed network, a destination node may receive the first bit of a packet before the local node has completely finished transmitting all the bits of that packet. In a high-speed network, on the other hand, there may be situations where most of the packets are in transit on links, that is they may not yet have reached the destination node even though many other packets have since been transmitted from the local node. The amount of such data in transit on links is characterized by the *bandwidth-delay product*, i.e., the propagation distance multiplied by the transmission rate. Since the only packets we can control are those the packets stored in nodes, high-speed networks contain many packets which cannot be controlled. In this situation, since the delay in applying control greatly affects the network performance, a very rapid control mechanism is required, in which the control delay should be as short as possible. So, in high-speed networks, the framework used for time-sensitive control is inevitably that of autonomous decentralized systems [2]–[4].

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 used in the control method.

- The control which is applied should take effect immediately.

There are many other papers which report studies on the optimization of flow control problems in a framework of solving linear programs [5]–[8]. 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 an extremely short time-scale. Since these control schemes are forms of centralized control, they cannot be adopted for high-speed networks.

Decentralized flow control by end hosts, including TCP, is widely used in current networks, and there is much research in this area [7]–[9]. However, since end-to-end or end-to-node control cannot be applied to decision-making in a time-scale shorter than the round-trip delay, it is inadequate for application to decision-making in an extremely 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, such a control delay greatly affects the network performance.

Bartal et al. [10] studied the global optimization of flow control using local information. The motivation for their work was to enable the distributed routers in high-speed networks to make appropriate decisions on flow control as quickly as possible, and they studied the problem in a framework of solving linear programs by means of distributed agents. Though this motivation is similar to that of our work, their study assumed that the distributed agents can obtain detailed information about networks if we allow sufficient time to gather it. As stated above, our standpoint is based on the fact that it is not possible to obtain detailed, useful, and up-to-date information about the whole network in a high-speed network environment.

In our previous studies, we investigated the behavior of local packet flows and the network-wide performance when a node is congested, and proposed DFC [1], [2]. In addition, we investigated the stability and adaptability of the network-wide performance when the network has inhomogeneous configurations with respect to link length [11]. DFC provides a framework in which the implementation of decision-making at each node leads to high performance for the whole network. We used a closed network model for the evaluation in this research because the number of packets in the network is unchanged in a closed network model and it permits comparison with other control mechanisms under the same conditions. To apply DFC to real networks, there are some technical issues to be overcome. Particularly important issues are:

- applying DFC to an end-to-end flow (an open network),
- compatibility and complementarity with other control mechanism (e.g. TCP), and
- applying DFC to multiple flows.

This paper discusses the application of DFC to an open network model. We identify the key issues in applying our flow control to an open network and show a solution for them.

2. Preliminary Description of Diffusion-Type Flow Control

2.1 Diffusion-Type Flow Control Mechanism

Figure 2 shows the interactions between nodes in our flow control method, using a network model with a simple 1-dimensional configuration, which represents a part of a network along a flow path.

All nodes have two incoming and two outgoing links, for a one-way packet stream and for feedback information, that is, node i ($i = 1, 2, \ldots, N$) transfers packets to node $i + 1$ and node $i + 1$ sends feedback information (consisting of information about node $i + 1$) to node i. For simplicity, we assume that all packets have a fixed length in bits.

All nodes are capable of receiving and sending feedback information. 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 the 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 $i + 1$. The framework for node behavior and flow control may be summarized as follows:

- When the feedback information (details are stated later) from the downstream node $i + 1$ is received, each node i autonomously determines the transmission rate, based only on information available to the node i itself. The available information is feedback information obtained from the downstream node $i+1$ and node i's own 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 the transmission rate (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 $i - 1$. The interval for generating feedback information is proportional to the propagation delay, d_{i-1}, between nodes $i - 1$ and i.

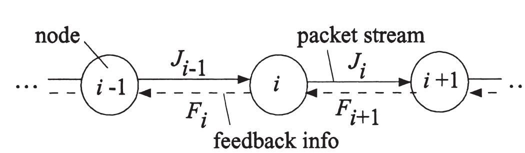

Fig. 2 Node interactions in our flow control model.

- 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, let us look at the details of DFC. The transmission rate $J_i(\alpha, t)$ of node i at time t is determined by

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

and

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

where $L_i(t)$ denotes the value of link capacity between node i and $i+1$ at time t, $n_i(t)$ denotes the number of packets stored 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 nodes i and $i+1$. In addition, $r_i(t - d_i)$ and $n_{i+1}(t - d_i)$ are notified from the downstream node $i+1$ with a propagation delay d_i. Parameter α (≥ 1) is a constant and is called the flow intensity multiplier. Parameter D_i (> 0) is chosen to be inversely proportional to the propagation delay [11] as follows:

$$D_i = D\,\frac{1}{d_i} \propto (d_i)^{-1}, \tag{3}$$

where D is a positive constant and is called the diffusion coefficient.

The feedback information, $\mathbf{F}_i(t)$, created by node i consists of two quantities as follows:

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

Node i notifies this to the upstream node $i-1$. Here, the target transmission rate is determined as

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

2.2 Principle of Diffusion-Type Flow Control

There are phenomena in nature where local interactions on the micro-scale produce symmetry on the macro-scale. For example, local interactions among water molecules lead to highly symmetrical snow flakes. By applying the proposed mechanism, although time-sensitive control on a short time scale can not make use of global information about the network, we may be able to control the network-wide performance through local decision making at the nodes.

In the framework of our control method, network-wide performance is controlled indirectly, but immediately, through the application of a predefined rule for local decision making at nodes. Nodes at distant locations do not communicate with each other. However, in spite of the constraint of the speed of light, appropriate control of network-wide performance may be achieved quickly. Since centralized control cannot implement performance control quickly due to the constraint of the speed of light, our control may actually appear to take effect faster than the speed of light.

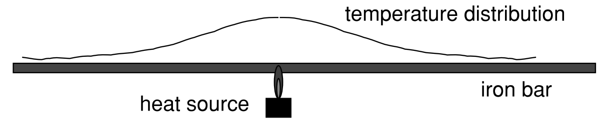

Fig. 3 Example of thermal diffusion phenomenon.

The principle of our flow control model can be explained through the following analogy [12], [13].

When we heat a point on a cold iron bar, the temperature distribution forms a normal distribution, and heat spreads through the whole bar as a diffusion phenomenon (Fig. 3). In this process, the action in a minute segment of the iron bar is very simple; heat flows from the higher temperature side towards the lower temperature 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 only on information available to it locally, the temperature distribution of the whole iron bar exhibits orderly behavior.

Our flow control is based on the diffusion-type equation and we expect that any congestion in the network will be dissipated with time, like diffusion phenomena. In our framework, by designing an appropriate rule for decision-making at nodes, we can get the flow control mechanism to work effectively as an autonomous decentralized system. The network-wide state of the network becomes the solution of a certain temporal evolution equation. Although the behavior of each node obeys a predefined rule and its decision is based only on local information, we can expect the performance of the whole network to exhibit an orderly pattern of behavior.

In DFC, each node controls its local packet flow to a locally-derived value; one term in the expression giving this value is proportional to the difference between the number of packets stored in the node and the number stored in the adjacent node and is inversely proportional to the distance between these nodes (refer to the second term in Eq. (2)). As a result the distribution of the number of packets stored in the different nodes in the network becomes uniform over time. Using this control mechanism, the state of the whole network is controlled indirectly through the autonomous action of each node.

Let us consider an example using continuous approximation for simplicity. After replacing i with x and $d_i \to 0$, the packet flow $J(x, t)$ in our framework may be written as

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

where $r(x, t)$ and $n(x, t)$ denote the drift component of the flow and the number of packets, respectively, at position x and time t. As each node handles the flow, it requires not full, global information but only local information relating to x and its immediate vicinity. So, autonomous decentralized control applies effectively to this framework. The temporal evolution of the number of packets at x may be written as

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

Equation (7) is obtained by combining the continuous equation $\partial n(x,t)/\partial t = -\partial J(x,t)/\partial x$ and Eq. (6). Equation (7) is a diffusion-type equation and it will lead to a diffusion phenomenon with respect to the number of packets.

3. Issues for Applying Diffusion-Type Flow Control to an Open Network Model

This section investigates the key issues involved in applying the DFC mechanism described in the previous section to an open network model. Figures 4 and 5 show the open network models, each with 60 nodes, which were used in the simulations. Although each 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. We represent the lengths of links by their delays, and the lengths of the individual links are determined randomly, in advance, as they follow a log-normal distribution with a mean of 1.0 [unit of time] and a variance of 5.2 [unit of time^2].

Our simulation scenario is to investigate the stability under congestion caused by the presence of a bottleneck link. The packet transmission rate of all links is $L_i = 100$ [packets/unit of time] except for the bottleneck link. We consider two cases, corresponding to different locations of the bottleneck link; in case A (Fig. 4) the bottleneck link is between nodes 30 and 31, and $L_{30} = 50$ [packets/unit of time], while in case B (Fig. 5) it is between nodes 1 and 2, and $L_1 = 50$ [packets/unit of time].

There are 5700 packets in the network at $t = 0$, and all the packets are distributed randomly on the links. We set the values of the parameters as $D = 0.1$ and $\alpha = 1.0$. At the ingress point of the network, the rate of packet flow is regulated by $r_0(t)$ which is specified by node 1.

Figures 6 and 7 show the results for cases A and B, respectively. The horizontal axis of each graph denotes node ID and the vertical axis denotes the number of packets stored in the node, i.e., the queue length in the node. In addition,

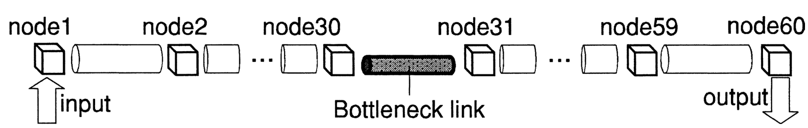

Fig. 4 Network configuration model (case A).

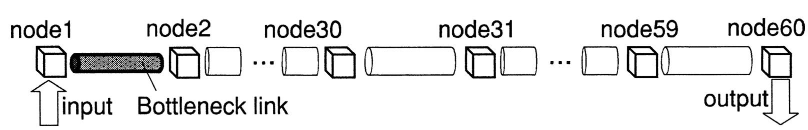

Fig. 5 Network configuration model (case B).

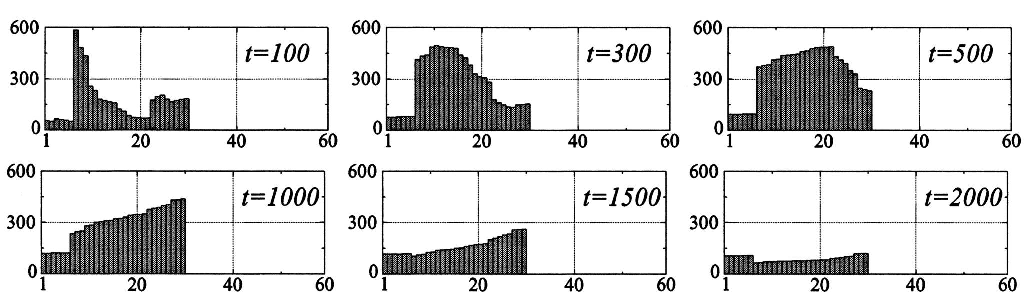

Fig. 6 Temporal evolution of distribution of packets stored in each node for Case A.

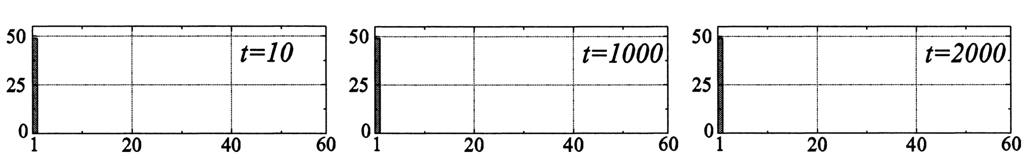

Fig. 7 Temporal evolution of distribution of packets stored in each node for Case B.

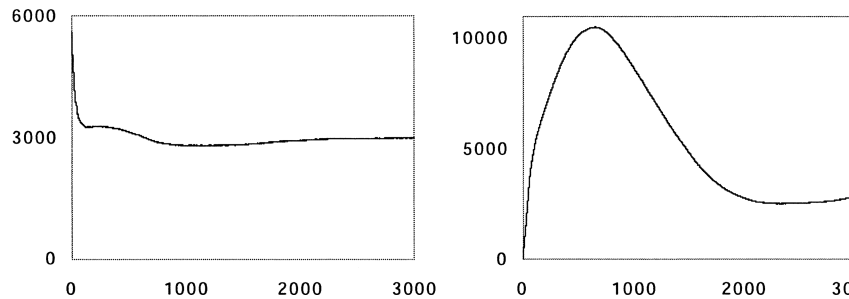

Fig. 8 Total numbers of packets in transit on links (left) and stored in nodes (right) for DFC described in Sect. 3.

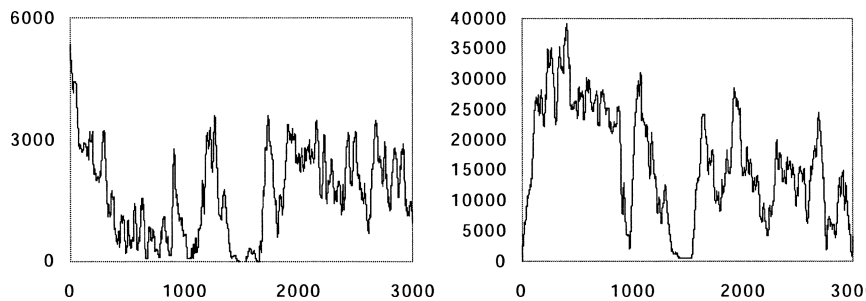

Fig. 9 Total numbers of packets in transit on links (left) and stored in nodes (right) for centralized control.

t denotes the simulation time and initially $t = 0$. In case A, where the initially congested node is node 30, the distribution of the number of packets stored in nodes becomes smoothly distributed over the part of the network before the bottleneck link, and the number of stored packets decreases with time. In case B, on the other hand, the number of packets stored in the congested node 1 does not decrease with time, and the distribution of the number of packets does not become uniform over time.

Next, we investigate the temporal evolution of the number of packets in transit on the links and stored in the nodes for case A in which the link between nodes 30 and 31 is the bottleneck link. The simulation conditions are same as before. The left and right sides of Fig. 8 show the results for the total number of packets in transit on links and the total number of packets stored in nodes in the network, respectively. The horizontal axes denote the simulation time and the vertical axes denote the number of packets. The total number of packets in transit on links indicates the transmission efficiency of the network, and we call this the total throughput. From the quantitative point of view, for the case where the link capacity of the bottleneck link $L_{30} = 50$, the maximum value of the sustainable total throughput (the maximum number of packets that can be transmitted stably on the links) is about 3000, i.e., 50 packets/link $\times$ 59 links. Thus, DFC achieves about 100% of the maximum value of the total throughput and its value is stable. On the other hand, the right side of Fig. 8 shows that there are many packets stored in nodes. A large number of packets stored in nodes is undesirable because it causes an increase in delay and packet loss.

These phenomena were not observed in the evaluation of the closed network model. We can see from these results that it is necessary to regulate the packet flow carefully, especially in the open network model in which the total number of packets in the network changes with time.

From the results mentioned above, the issues to be considered when applying DFC to an open network model may be summarized as follows:

- The nature of the boundary condition for determining the packet rate at the ingress node in order to the number of packets to diffuse smoothly over the network.
- Traffic shaping to prevent excessive traffic input from the ingress node.

These will be discussed later.

At this point, to demonstrate a feature of DFC as an autonomous decentralized system, we may compare Fig. 8 with results obtained from a centralized flow control model under the same simulation conditions. For centralized control, we consider the following model. Each node sends its feedback information $\mathbf{F}_i(t)$ to the source node. The source node calculates the packet transmission rate $J_i(\alpha, t)$ for each node from the collected feedback information and sends it back to the corresponding node. Both the collection of $\mathbf{F}_i(t)$ and the distribution of $J_i(\alpha, t)$ incur propagation delay. Figure 9 shows the results of the total number of packets in transit on links and the total number of packets stored in nodes, obtained using from centralized control. The total number of packets in transit on links is unstable and many packets are stored in nodes. This comparison indicates that it is inappropriate to apply such a centralized control mechanism to high-speed networks.

4. Diffusion-Type Flow Control Model Adapted for Open Networks

4.1 Packet Rate at the Boundary Nodes

In this subsection, we consider the boundary condition of the rule for determining the packet rate in DFC, which applies at the ingress to the network.

Here we consider the situation where nodes in other networks and/or end hosts 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 is able to adjust the packet rate to the requested rate notified from the downstream node. That is, an external node 0 cannot calculate the packet rate $J_0(\alpha, t)$ using Eq. (2), but can adjust its packet 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 stored packets in the external 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,

$$r_0(t) := \tilde{J}_0(\alpha, t + d_0) = \alpha\, J_1(1, t) - D_0\, n_1(t). \tag{8}$$

This quantity can be calculated from information that is available for node 1.

4.2 Packet Shaping

In this subsection, we consider two extensions to the DFC model in order to regulate the packet flow at the ingress of the network.

In the first model, all nodes regulate the packet flow rate so that it is less than or equal to the minimum value of link capacity of all the downstream links. In contrast, in the second model, only the ingress node uses its minimum value when calculating the packet flow rate.

(1) Model 1

When node i receives feedback information from the downstream node $i + 1$, it determines the transmission rate for packets to the downstream node $i + 1$, and adjusts its transmission rate towards the downstream node $i + 1$. Then, the packet transmission rate is determined as

$$J_i(\alpha, t) = \max(0, \min(\ell_i(t), \tilde{J}_i(\alpha, t))), \tag{9}$$

where $\ell_i(t)$ denotes information about the maximum value of available capacity. In addition, node i generates feedback information $\mathbf{F}_i(t)$ as

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

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

The feedback information is determined as

$$r_{i-1}(t) = \max(0, \min(\ell_i(t), \tilde{J}_i(1, t))), \quad \text{and} \tag{11}$$
$$\ell_i(t) = \min(L_i, \ell_{i+1}(t - d_i)). \tag{12}$$

Note that the calculation of $r_{i-1}(t)$ uses $\alpha = 1$, and $\ell_i(t)$ means the minimum value of link capacity of all the downstream links. A remarkable feature of Model 1 is that the maximum values of $J_i(\alpha, t)$ and $r_{i-1}(t)$ are bounded by $\ell_i(t)$.

(2) Model 2

Node i's packet transmission rate to the downstream node $i + 1$ is determined as

$$J_i(\alpha, t) = \max(0, \min(L_i(t), \tilde{J}_i(\alpha, t))). \tag{13}$$

In addition, node i generates feedback information $\mathbf{F}_i(t)$ as

$$\mathbf{F}_i(t) = (r_{i-1}(t), n_i(t), \ell_i(t)), \tag{14}$$

and notifies this information to the upstream node $i - 1$. The feedback information is expressed as:

$$r_{i-1}(t) = J_i(1, t), \quad \text{and} \tag{15}$$
$$\ell_i(t) = \min(L_i, \ell_{i+1}(t - d_i)). \tag{16}$$

The difference between Models 1 and 2 is that the maximum values of $J_i(\alpha, t)$ and $r_{i-1}(t)$ are bounded by L_i in Model 2, while they are bounded by $\ell_i(t)$ in Model 1.

Both flow control models calculate the transmission rate $J_0(\alpha, t)$ for packets to flow at the ingress to the network as follows:

$$J_0(\alpha, t) = \min(\ell_0(t), r_0(t)). \tag{17}$$

4.3 Evaluation Using the Extended Flow Control Models

First, we examined the effectiveness of the boundary condition described in Sect. 4.1. We chose the parameter values as $D = 0.1$ and $\alpha = 1.0$. Figure 10 shows the temporal evolution of the distribution of packets stored in each node with respect to DFC with the boundary condition applied. The network model is case B, where the bottleneck link is between nodes 1 and 2. This figure shows the packet distribution is diffused appropriately even if the ingress node is congested.

Next, we compared the network performance of the DFC Models 1 and 2 described in Sect. 4.2. The network model used is case A.

Figures 11 and 12 show the results for the flow control Models 1 and 2, where the horizontal axes denote the simulation time and the vertical axes denote the total number of packets in transit on links or stored in nodes.

We can see from Figs. 11 and 12 that the total numbers of packets in transit on links becomes stable in both models. However, there is a difference with regard to the number of packets stored in nodes. In Model 1, the total number of packets stored in nodes is smaller than in DFC described in Sect. 3 and, while it becomes stable, it does not decrease below 3000. In Model 2, on the other hand, the total number

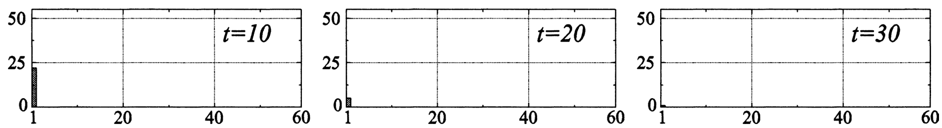

Fig. 10 Temporal evolution of distribution of packets stored in each node for DFC using the boundary condition.

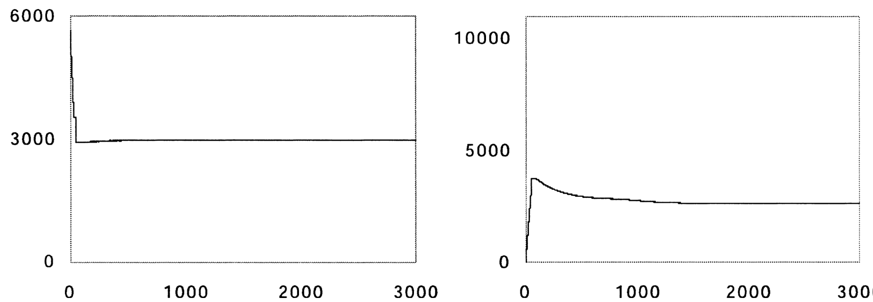

Fig. 11 Total numbers of packets in transit on links (left) and stored in nodes (right) for DFC (Model 1).

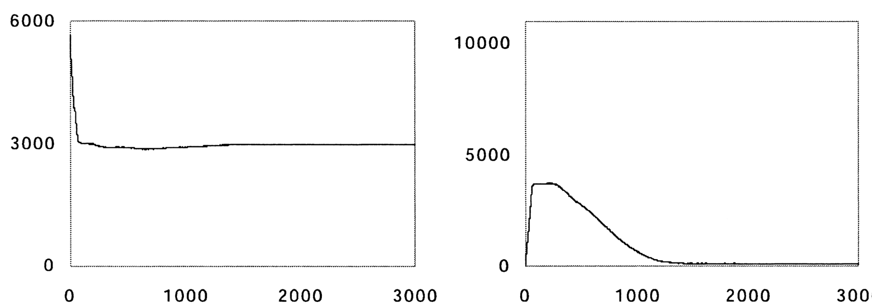

Fig. 12 Total numbers of packets in transit on links (left) and stored in nodes (right) for DFC (Model 2).

of packets stored in nodes decreases rapidly and falls to zero with time.

Next, we compared the behavior of the packet distribution among individual nodes, in order to investigate the performance of Models 1 and 2. Figures 13 and 14 show the simulation results for Model 1 and Model 2, respectively. The horizontal axis of each graph denotes the node ID, the vertical axis denotes the number of packets stored in the node, and t denotes the simulation time (initially $t = 0$).

We can see from Fig. 13 that in Model 1 the number of packets stored in each node becomes smoothly distributed over the network but a number of stored packets remain in the network. For Model 2, on the other hand, the number of packets stored in nodes becomes smoothly distributed over the network and decreases with time.

This difference arises because in Model 1 all nodes calculate the packet transmission rates using the minimum value of link capacity of all the downstream links. This restriction, which is imposed on calculating the rate in Model 1, is too severe. So, the transmission rates calculated in Model 1 differ greatly from the ideal rates, which are determined by Eq. (2) and govern the diffusion phenomenon. Our results also show that the rule for determining the packet rate at a node, which is the most important feature of our flow control, is crucially important for the stability and adaptability of the whole network performance.

5. Conclusions

In this paper, we have identified the key issues associated with the application of DFC to an open network model, and demonstrated a solution. The first issue relates to the boundary condition for determining the packet rate. By introducing an assumption that an external node has no stored packets, we can have an appropriate boundary condition that enables smooth diffusion of packet distribution even if the ingress node is congested. The second issue is the shaping mechanism required to control the number of packets in the network appropriately. We have studied two different shaping mechanisms, Models 1 and 2, and compared them. Our results show that the shaping mechanism of Model 2 not only provides about 100% link utilization but also results in only a small number of packets stored in nodes.

The two issues for applying DFC to the open network model have been solved by introducing two mechanisms: the boundary condition (described in Sect. 4.1) and packet

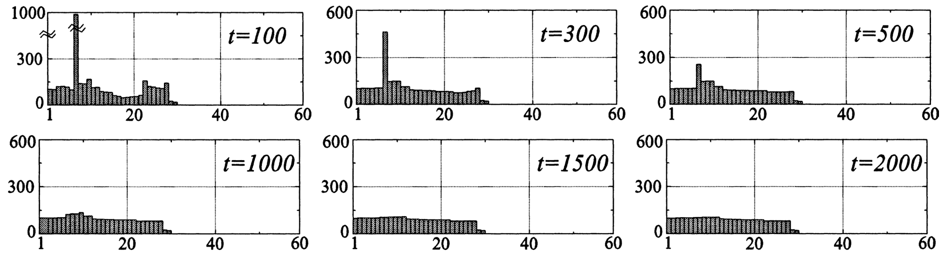

Fig. 13 Temporal evolution of distribution of packets stored in each node (Model 1).

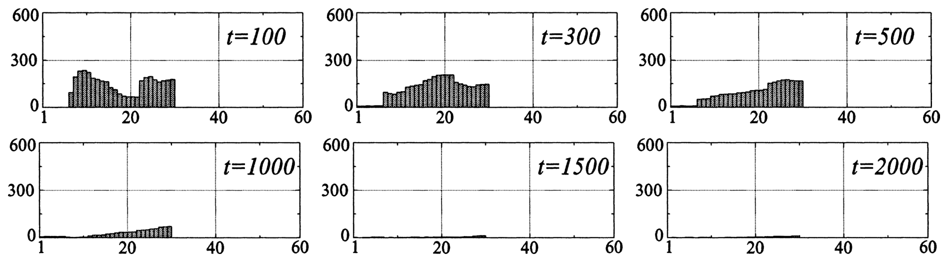

Fig. 14 Temporal evolution of distribution of packets stored in each node (Model 2).

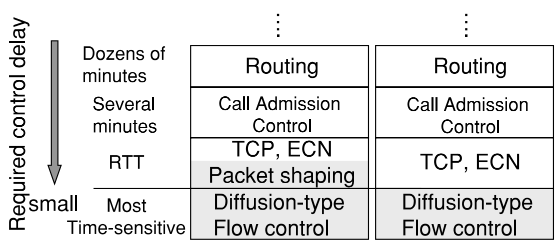

Fig. 15 Classification of various control mechanisms with respect to their effective timescales, with and without packet shaping.

shaping (described in Sect. 4.2). Finally, we have restructured the relationships between the proposed two mechanisms and the previous DFC (described in Sect. 3).

Various controls in the network can be classified from the point of view of their particular time-scale of control operation. Figure 15 shows a comparison of different types of control according to such a classification. For example, routing and call admission control fall into the long and medium time scales, respectively. Individual control mechanisms work well for their appropriate time-scales and they cooperate with each other. Window flow control such as TCP acts in the time scale of the round-trip delay. In high-speed networks, since a lot of packets are being transmitted on links, control delay greatly affects the performance. Our target is a time-scale shorter than the RTT. The left and right hand sides of Fig. 15 show appropriate classification, with and without packet shaping, respectively. Determination of the packet rate at the boundary node, based only on the information available for the node, can be applied to decision-making in a time-scale as short as that achieved by DFC. Moreover, this application of this rule is essential in achieving appropriate diffusion of packet density. Therefore, the boundary condition for determining the packet rate should be included in the framework of DFC.

The packet shaping mechanism requires knowledge of the minimum link bandwidth, $\ell_0(t)$, along a flow. Since this requires global information about the network, it does not sit comfortably in the framework of DFC, which is based only on local network information. Consequently, the packet shaping mechanism should be classified into a higher layer. If other control mechanisms (e.g. TCP) are applied and they regulate the number of packets in the network appropriately, applying packet shaping is not mandatory and we expect the combination of DFC and other control mechanisms also to work well.

This paper has not addressed the use of TCP flow control for the end-to-end flow, and we should verify the compatibility and complementarity of DFC with TCP or other higher-layer protocols. In addition, another residual issue for applying DFC to real networks is its applicability to multiple flows. This paper has focused on issues resulting from a single flow, but we should take the existence of other traffic into consideration. Combining fair queueing mechanism with DFC might be a promising approach to solve this issue.

The issues are areas for further study in extending the application of DFC.

References

[1] 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.

[2] 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.

[3] M. Aida and K. Horikawa, "Stability analysis for global performance of flow control in high-speed networks based on statistical physics," IEICE Trans. Commun., vol.E82-B, no.12, pp.2095–2106, Dec. 1999.

[4] K. Horikawa, M. Aida, and T. Sugawara, "Traffic control scheme under the communication delay of high-speed networks," Proc.

Int. Conf. on Multi-Agent Systems (ICMAS'96), pp.111–117, Dec. 1996.
[5] 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.
[6] 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.
[7] 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.
[8] S. Kunniyur and R. Srikant, "A decentralized adaptive ECN marking algorithm," Proc. IEEE GLOBECOM'00, pp.1719–1723, 2000.
[9] 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.
[10] 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.
[11] C. Takano and 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.
[12] M. Aida and C. Takano, "Principle of autonomous decentralized flow control and layered structure of network control with respect to time scales," Supplement of the ISADS 2003 Conference Fast Abstracts, pp.3–4, Pisa, Italy, April, 2003.
[13] M. Aida and C. Takano, "Principle of diffusion-type autonomous decentralized flow control," Proc. Commun. Conf. IEICE 2003, B-7-13, 2003.

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

Masaki Aida received his B.Sc. and M.Sc. 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 the 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 IEICE Young Investigators Award in 1996. He is a member of the IEEE and the Operations Research Society of Japan.