Diffusion-Type Flow Control Scheme for Multiple Flows

Masaki Aida, Chisa Takano and Akira Miura

Faculty of System Design, Tokyo Metropolitan University
Hino-shi 191-0065, Japan
aida@m.ieice.org

Traffic Research Center, NTT Advanced Technology Corporation
Musashino-shi 180-0006, Japan
chisa@m.ieice.org

Network Laboratories, NTT DoCoMo Inc.
Yokosuka-shi 239-8536, Japan
miura@netlab.nttdocomo.co.jp

Abstract: 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 on 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 concentrated on flow control for a single flow. In this paper, we propose the diffusion-type flow control mechanism for multiple flows. This scheme enables us to recover quickly from congestion and achieve fairness among flows.

1. INTRODUCTION

The rapid spread of the Internet will necessitate the construction of higher-speed backbone networks 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 the state of a node varies rapidly in accordance with its processing speed although the propagation delay is constant. 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.

There are many other papers reporting studies on flow control optimization in a framework of solving linear programs [1,2,3]. These studies assume the collection of global information.
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 time-scale. Therefore, 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 a lot of research in this area [3,4]. However, since end-to-end or end-to-node control cannot be applied to decision-making on a time-scale shorter than the round-trip delay, it is inadequate for application to support decision-making on a very short time-scale. In low-speed networks, a control delay on 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 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. To achieve rapid control on a shorter time scale than the RTT, it is preferable to apply control by the nodes rather than by the end hosts.

We 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 apply quick decision-making. Although decision-making at a local node should lead to action suitable for the local performance of the networks, the action is not guaranteed to be appropriate for the overall network-wide performance. Therefore, the implementation of decision-making at each node cannot lead to optimum performance for the whole network.

In our previous studies, we proposed a diffusion-type flow control (DFC) [5,6]. DFC provides a framework in which the implementation of the decision-making of each node leads to high performance for the whole network. The principle of our flow control model can be explained through the following analogy [7].

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 work on DFC focused on flow control for a single flow and the packet density was uniformized along a one-dimensional path of the flow [7]. So, it was not directly related to the uniformization of packet density over the whole network. In this paper, we focus on an environment with multiple flows and we reconsider the packet density uniformization to clarify our goal. We propose DFC for multiple flows, which achieves packet density uniformization beyond the one-dimensional path of the flow. The effective characteristics of the proposed control are shown through simulations.

2. PRELIMINARY DESCRIPTION OF CONTROL FOR A SINGLE FLOW

In the case of Internet-based networks, to guarantee end-to-end quality of service (QoS) of a flow, a 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.

Our flow control for a single flow consists of two parts: the main flow control part and an optional traffic regulation part. The main part is DFC and it works to uniformize the number of packets in routers along a path of the flow. The optional part regulates the volume of traffic at the ingress point of the network. This function is not mandatory but can be replaced by another flow control, e.g., the window control of TCP.

2.1 DIFFUSION-TYPE FLOW CONTROL

Figure 2 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 to node \( i \). For simplicity, we assume that packets have a fixed length in bits.

All nodes are capable of receiving feedback information from adjacent downstream nodes, and sending it to 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 \( i + 1 \). The framework for node behavior and flow control is summarized as follows:

- Each node \( i \) autonomously determines the transmission rate \( J_i \) based only on information directly available to it, that is, the feedback information obtained from the downstream node \( i + 1 \) and its own feedback information.

- 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 with a fixed interval \( \tau_i \).

- The above rules are 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 explain the details of DFC. The transmission rate \( J_i(\alpha, t) \) of node \( i \) at time \( t \) is determined by

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

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 nodes \( i \) and \( i + 1 \).

In addition, \( r_i(t - d_i) \) and \( n_{i+1}(t - d_i) \) are reported every fixed period \( \tau_{i+1} \) from the downstream node \( i + 1 \) with propagation delay \( d_i \). Parameter \( \alpha \) (\( \geq 1 \)), which is a constant, is the flow intensity multiplier. Parameter \( D_i \) is chosen to be inversely proportional to the propagation delay [6] as \( D_i = D/d_i \), where \( D \) (\( > 0 \)), which is a positive constant, is the diffusion coefficient.

The feedback information \( F_i(t) \) created every fixed period \( \tau_i \) by node \( i \) consists of the following two quantities: \( F_i(t) = (r_{i-1}(t), n_i(t)) \). 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 \( r_{i-1}(t) = J_i(1, t) \). Moreover, the packet flow \( J_i(t) \) 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 packet flow (1) is expressed as

\[
\tilde{J}(\alpha, x, t) = \alpha r(x, t) - D \frac{\partial n(x, t)}{\partial x},
\]

and the temporal variation of the packet density \( n(x, t) \) is represented by a diffusion-type equation,

\[
\frac{\partial n(x, t)}{\partial t} = -\alpha \frac{\partial r(x, t)}{\partial x} + D \frac{\partial^2 n(x, t)}{\partial x^2},
\]
using the continuous equation $\partial n(x,t)/\partial t = -\partial \tilde{J}(\alpha, x, t)/\partial x$. That is, 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 to the requested rate reported by the downstream node. That is, an external node 0 cannot calculate the transmission rate $J_0(\alpha, t)$ using (1), but can adjust its transmission rate to $r_0(t - d_0)$, which was reported by 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 that the number of packets stored in the other networks’ node is 0. The target rate $r_0(t)$, reported by node 1, is created as $\tilde{J}_0(\alpha, t)$ with the above assumption. That is, $r_0(t) := J_0(\alpha, t + d_0) = \alpha J_1(1, t) - D_0 n_1(t)$. This quantity can be calculated just from information known to node 1.

### 2.2 Packet-Rate Regularization at the Ingress

Although DFC excels in quick uniformization of packet density, there are few effects that prevent an excessive number of packets flowing in from the outside the network because DFC is based on autonomous operation of the node using local information, so we should 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 keeps the transmission rate less than or equal to the minimum value of the available link capacity among all the downstream links [7].

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 by (1). In addition, node $i$ generates feedback information $F_i(t)$ as $F_i(t) = (r_{i-1}(t), n_i(t), \ell_i(t))$, and reports this information to the upstream node $i - 1$. The feedback information is expressed as $r_{i-1}(t) = J_i(1, t)$, and $\ell_i(t) = \min(L_i, \ell_{i+1}(t - d_i))$.

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

$$J_0(\alpha, t) = \max(0, \min(\ell_0(t), r_0(t - d_0))) = \max(0, \min(\ell_0(t), \tilde{J}_0(\alpha, t))). \quad (4)$$

Note that $\ell_0(t)$ is calculated from $\ell_1(t - d_0)$ reported by node 1 and from the bandwidth of access link $L_0$ (its propagation delay is $d_0$). Therefore, it is not necessary to calculate (4) at a node outside the network. If the node outside the network does not support DFC, it cannot calculate (4). In this case, node 1 reports $\min(\ell_1(t), r_0(t))$ as the shaping rate
to the node outside of the network. Then, at least, the node outside of the network can regulate the input traffic according to \( \min(\ell_1(t), r_0(t)) \) reported by node 1.

### 3. CLASSIFICATION OF PACKET DENSITY UNIFORMIZATION

In this section, we reconsider the uniformization of packet density and classify its methods.

(a) **Serial Diffusion**: To avoid packet losses, the number of packets of the target flow in routers along the flow’s path is uniformized. This type is further divided into two types; (a-1) Backward Serial Diffusion: with respect to a bottleneck link, the number of packets is uniformized toward the upstream direction. (a-2) Forward Serial Diffusion: with respect to a bottleneck link, the number of packets is uniformized toward 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 uniformized among all the flows.

In our previous work, our target was serial diffusion, in particular, backward serial diffusion, because, for a single flow environment, the bottleneck link with bandwidth \( L_i \) prevents the packet flow at node \( i \) being greater than \( L_i \) (see Sec. 5.2). However, in a multiple flow environment, the bandwidth of the bottleneck link is shared by multiple flows, so some flows may have larger rates than others. For fast congestion recovery, both backward and forward serial diffusion are important. We can expect both to be achieved by appropriately setting the available bandwidth of each flow.

We hope that the effect of packet uniformization will spread not only along the path of the target flows but also to the whole network, so packet uniformization among flows by parallel diffusion is important. To achieve parallel diffusion, we assign an appropriate available bandwidth \( L_i \) for each flow passing through node \( i \).

From the above considerations, appropriate setting of \( L_i \) is the key issue to achieve DFC with backward and forward diffusion and parallel diffusion.

### 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 link bandwidth fairly. Extension to the case where flows require different bandwidths is easy.

There are \( M_i \) flows sharing the link between nodes \( i \) 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: \( n_j^i(t) \): the number of packets of flow \( j \) in node \( i \) at time \( t \), \( r_j^i(t - d_i) \): the rate for flow \( j \) reported by feedback information from the downstream node \( i + 1 \), \( n_j^{i+1}(t - d_i) \): the number of packets of flow \( j \) in node \( i + 1 \) reported by feedback information from the downstream node \( i + 1 \), and \( L_j^i(t) \): the available bandwidth for flow \( j \) of a link from node \( i \) to \( i + 1 \).

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 $J_i^t(\alpha, t)$ on the basis of only the local information directly available to it. In addition, each node $i$ autonomously creates feedback information $F_i^j(t)$ 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 generations of feedback information be $M_i(t)$. That is, $M_i(t)$ is the average number of distinct flows of packets in node $i$. In general, $M_i(t) \leq M_i$. Let the link bandwidth from node $i$ to $i + 1$ be $B_i$. Then, transmission rate for flow $j$ is determined as $\tilde{J}_i^j(\alpha, t) = \max(0, \min(L_i^j(t), \tilde{J}_i^j(\alpha, t)))$, and

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

On the other hand, feedback information for flow $j$ generated by node $i$ consists of $F_i^j(t) = (r_{i-1}^j(t), n_i^j(t))$, reported to the upstream node $i - 1$. Here, the target rate for flow $j$ is $r_{i-1}^j(t) = \max(0, \min(B_i/\bar{M}_i(t), \tilde{J}_i^j(1, t)))$. When node $i$ is at the ingress of the network ($i = 1$), node 1 reports $r_0^1(t) = \alpha J_1^1(1, t) - D_0 n_1^1(t) = \tilde{J}_0^1(\alpha, t + d_0)$ to the upstream node or end host (outside of the network). If packet shaping at the ingress of the network is required, we can apply a similar way described in Sec. 2.2.

4.2 DETERMINATION OF AVAILABLE BANDWIDTH

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

Let the bandwidth of the link from node $i$ to $i + 1$ be $B_i$. Then, $L_i^j(t) (j = 1, 2, \ldots, M_i)$ must satisfy $\sum_{j=1}^{M_i} L_i^j \leq B_i$. If we choose $L_i^j(t)$ as a fixed value equal to $B_i/M_i$, interference among flows does not occur and we do not get both forward serial diffusion and parallel diffusion. This is because 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 its available bandwidth. In particular, if there are a lot of flows, the ideal transmission rate $\tilde{J}_i^j(\alpha, t)$ is frequently $\sum_{j=1}^{M_i} \tilde{J}_i^j(\alpha, t) > B_i$, and the probability of each flow getting its ideal transmission rate $\tilde{J}_i^j(\alpha, t)$ is low. This prevents the smooth uniformization of the packet density. So, we take into consideration two conditions: relative value of the ideal transmission rate of each flow and $\sum_{j=1}^{M_i} J_i^j(\alpha, t) \leq B_i$. Then, the simplest way to determine $L_i^j$ is to assume that the bandwidth $B_i$ is shared by flow with a weight $\tilde{J}_i^j(\alpha, t)$, that is,

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

This rule means that a flow with larger $\tilde{J}_i^j(\alpha, t)$ 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. We can expect this to achieve both forward serial diffusion and parallel diffusion.
5. SIMULATION RESULTS

5.1 EVALUATION MODEL

To demonstrate the characteristics of DFC, this section compares simulation results of DFCs for multiple flows and for a single flow. In our evaluation, we used packet shaping with $\ell(t)$, and the DFC parameters were set as $D = 0.1$ and $\alpha = 1$, and the interval of the feedback information $F_i(t)$ was set to $d_{i-1}$.

Figure 3 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 the target end-to-end flow extracted from the whole network. We represent the lengths of links by their delays, and the length of links is 1.0 [unit time]. A packet has a fixed length and the link bandwidth is 100 packets per unit time.

The simulation scenario was as follows. There were two flows. Flow 1 began at time $t = 1000$ and background flow 2 began at time $t = 0$. The path of flow 1 was from node 0 to 60 and that of background flow 2 was from node 30 to 60. The maximum rate (without traffic regulation) of both flows was 100 packets per unit time (same as link bandwidth). After the flow 1 traffic entered the network, the link from node 30 to 31 became a bottleneck, and traffic of both flows was regulated by predefined rules for DFC and packet shaping. After congestion occurred, we investigated the temporal evolution of the network state, for DFC for both multiple flows and a single flow.

5.2 SIMULATION RESULTS FOR SERIAL DIFFUSION

The upper panel of Fig. 4 shows the temporal evolution of the total number of packets stored in each node, for flow 1, when DFC for a single flow was applied. We chose the available bandwidth at the bottleneck link as $L_{20} = L_{30} = B_{30}/2$. Because the transmission rate $J_{30}(1, t)$ was regulated to less than or equal to $B_{30}/2$ at node 30, the nodes downstream from node 30 were not congested. The effect of backward serial diffusion caused congestion at node 30 to spread to the upstream nodes and prevented packet losses at node 30. If we had not applied DFC, all stored packets would have concentrated at node 30, which might have caused packet losses.

The lower panel of Fig. 4 shows the result in the case where DFC for multiple flow was applied. The available bandwidth at the bottleneck link was determined by (6), and its value dynamically changed depending on the situation. This result reveals both backward and forward serial diffusion. We can also see that the time taken to recover

![Network model diagram](network_model.jpg)

Figure 3: Network model.
Results from DFC for a single flow.

Results from DFC for multiple flows.

Figure 4: Temporal evolution of the number of packets stored in nodes controled by DFC.

Figure 5: Temporal evolution of the total number of packets in transit on links for background flow 2 (left) and for flow 1 (right).

from congestion was shorter than in the case of a single flow.

Forward serial diffusion results from the available bandwidth of the background flow 2 being temporarily set to $L_{30} < B_{30}/2$. Thus, we should investigate the influence on flow 2. We should investigate both the number of stored packets and the volume of packet transmission of flow 2. The former evaluation is shown in the next subsection for parallel diffusion, and we show results of the latter case here. The volume of packet transmission at time $t$ can be denoted by the number of packets in transit on links of the network. Figure 5 shows the results. Since the maximum number of packets in transit on a link at a moment was 100, and flow 2 passed through about half of links of flow 1, the maximum total number of flow 2’s packets in transit on links was 3000 when $t \leq 1000$ and was 1500 when $t > 1000$. On the other hand, the maximum total number of flow 1’s packets in transit on links was 3000 after $t = 1000$. From Fig. 5, 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.

5.3 EVALUATION OF PARALLEL DIFFUSION

This subsection shows the temporal evolution of the number of each flow’s packets stored in the common node. Network model used in this evaluation was the same as in Sec. 5.1. The simulation scenario was as follows. There were three flows. Flow 1 began
at time $t = 0$ and background flows 2 and 3 began at time $t = 1000$. The path of flow 1 was from node 1 to 60, and the paths of background flows 2 and 3 were from node 30 to 60. The maximum rate (without traffic regulation) of flow 1 was 100 packets per unit time (same as link bandwidth), and each maximum rate of background flow 2 or 3 was 20 packets per unit time. After the traffic of background flows 2 and 3 entered the network, the link from node 30 to 31 became a bottleneck. Then the traffic was regulated by the predefined rules of DFC and packet shaping.

Figure 6 shows the temporal evolution of the number of each flow’s packets stored in the congested node 30. We can see that the number of flow 1’s packets is large. This is because the distance between the ingress node of flow 1 and node 30 was longer than that of other flows. After that, the number of packets for flow 1 decreased and was uniformized among other flows. If we had used DFC for a single flow with $L_{30}^1 = L_{30}^2 = L_{30}^3 = B_{30}/3$, the numbers of stored packets for background flows 2 and 3 would have been 0. Figure 6 implies that DFC for multiple flows achieves fast recovery from congestion by influencing the background flows 2 and 3. However, since the numbers of their flow’s packets stored in node 30 were sufficiently small, the influence on the background flows was small.

6. CONCLUSIONS

In this paper, we extended DFC to be applicable to multiple flows. DFC aims to prevent packet concentration at the congested node and to avoid packet losses. Uniformization of the number of packets is classified into two types: serial diffusion along the path of the flow and parallel diffusion among different flows. Our previous work on DFC treated only a single flow and it achieved only partial serial diffusion (backward serial diffusion). By choosing the available bandwidth for each flow appropriately, DFC for multiple flows achieves not only backward serial diffusion, but also forward serial diffusion and parallel diffusion. In addition, the proposed flow control can shorten the time taken to recover from congestion.

REFERENCES