Effective Delay-Controlled Load Distribution over Multipath Networks

Sumet Prabhavat, *Student Member*, *IEEE*, Hiroki Nishiyama, *Member*, *IEEE*,
Nirwan Ansari, *Fellow*, *IEEE*, and Nei Kato, *Senior Member*, *IEEE*

Abstract—Owing to the heterogeneity and high degree of connectivity of various networks, there likely exist multiple available paths between a source and a destination. An effective model of delay-controlled load distribution becomes essential to efficiently utilize such parallel paths for multimedia data transmission and real-time applications, which are commonly known to be sensitive to packet delay, packet delay variation, and packet reordering. Recent research on load distribution has focused on load balancing efficiency, bandwidth utilization, and packet order preservation; however, a majority of the solutions do not address delay-related issues. This paper proposes a new load distribution model aiming to minimize the difference among end-to-end delays, thereby reducing packet delay variation and risk of packet reordering without additional network overhead. In general, the lower the risk of packet reordering, the smaller the delay induced by the packet reordering recovery process, i.e., extra delay induced by the packet reordering recovery process is expected to decrease. Therefore, our model can reduce not only the end-to-end delay but also the packet reordering recovery time. Finally, our proposed model is shown to outperform other existing models, via analysis and simulations.

Index Terms—Delay minimization, load distribution, multipath forwarding, packet reordering, packet delay variation.

1 Introduction

The demand for network infrastructure in providing high-speed broadband network services that can support multimedia and real-time applications has been the major driving force for innovation and development of various networking technologies. Network capacity provisioning and Quality of Service (QoS) guarantees are key issues in fulfilling this demand. The heterogeneity and high degree of connectivity of various networks result in potentially multiple paths in establishing network connections. The exploitation of these multiple paths no longer aims only at circumventing single point of failure scenarios, but also focuses on facilitating network provisioning for multimedia data transmission and real-time applications, where its effectiveness is indeed essential to maximize high quality network services and guarantee QoS at high data rates [1], [2]. Bandwidth aggregation and network-load balancing are two major issues that have attracted tremendous amount of research, and a number of load distribution approaches have been proposed and studied [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], which will be briefly described later in the next section.

Multipath configurations can be established in several ways. For examples, a source node can distribute load via multiple next hops, emerging wireless technologies allow routes formed between a source and a network proxy via multiple wireless connections, and traffic flows from several sources are aggregated at and distributed by a gateway. Incorporating multiple physical/logical interfaces with a multipath routing protocol allows users to use multiple paths in establishing simultaneous connections [2], [3], [4], [16], [17], [18], [19], [20], [21], [22], [23], [24]. Devices must be equipped to perform traffic forwarding, which splits traffic into multiple paths as illustrated in Fig. 1. The traffic splitting component splits the input traffic into single packets or flows, each of which independently takes a path determined by the path selection component. If the forwarding processor, which is responsible for transmitting packets, is busy, it will be queued in the corresponding input queue. The bandwidth of a path is considered as the service rate of the forwarding processor which connects to the path. Network load caused by input traffic with arrival rate λ is shared among the multiple paths, i.e., the load of path p is assigned the traffic rate $\lambda_p \leq \lambda$. Therefore, bandwidth demand on each of multiple outgoing paths is likely to be smaller than that on the single outgoing path, as shown in Fig. 1.

Inefficient load distribution can cause many problems, e.g., load imbalance and packet reordering. The load imbalance problem can occur when the load is assigned on each path improperly with respect to the capacity of the path in terms of bandwidth and buffer size [8], [9], [25], [26]. If determination of a path takes into account of the queue length or level of path utilization, such system can achieve work-conserving load sharing [27] and can mitigate the load imbalance problem. Leaving at least one path to be idle (i.e., no load), while the other paths are busy, causes inefficient bandwidth utilization. The packet reordering problem also has a significant impact on the end-to-end performance perceived by users [28], [29], [30], [31], [32], [33], [34] and,

- *S. Prabhavat, H. Nishiyama, and N. Kato are with the Kato Laboratory, No. 3 Building of ECEI, Graduate School of Information Sciences, Tohoku University, Aramaki-Aoba 6-3-09, Aoba-ku, Sendai, Miyagi 980-8579, Japan.*
  E-mail: {sumetp, bigtree, kato}@it.ecei.tohoku.ac.jp, kpsumet@kmitl.ac.th.
- *N. Ansari is with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982. E-mail: Nirwan.Ansari@njit.edu.*

Manuscript received 28 June 2010; revised 13 Oct. 2010; accepted 3 Nov. 2010; published online 19 Jan. 2011.
Recommended for acceptance by A. Nayak.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-06-0382.
Digital Object Identifier no. 10.1109/TPDS.2011.43.

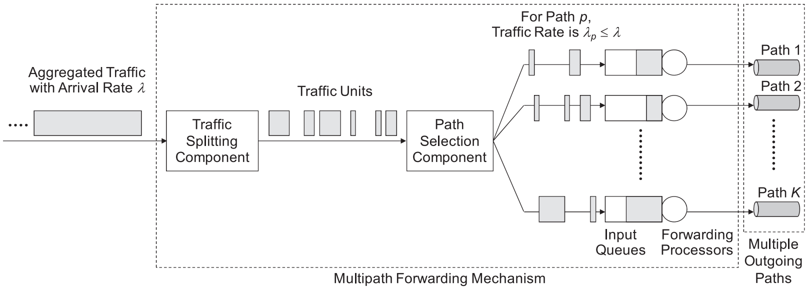


Fig. 1. Functional components of the multipath forwarding mechanism.

reportedly, is not a sporadic event if there is no mechanism to maintain packet ordering [34], [35], [36], [37]; it is likely to increase in a network with a large degree of parallelism. Packets arrived earlier have to wait for late packets in reordering buffers at the receiving destination. If late packets arrive within a receive timeout period, the transmission is successful; however, the waiting time causes packet delay. Otherwise, the late packet is treated as a lost one. In this paper, with the assumption that reordering buffer is infinitely large and that there is no timeout in waiting for late packets, the packet reordering problem causes additional delay without packet loss. The increase of the probability that the current packet takes a different path (from a previous one heading for the same destination), which has a different delay, leads to a higher degree of packet reordering [30], [31], [38], thus resulting in the extra delay.

Inefficient load distribution can degrade network performance as a result of a large variation of latency and a large latency to successfully transmitting a packet. The latency in the focus of this paper is the end-to-end delay in transmitting a packet and the additional time required in reordering the packet. End-to-end delay is the time it takes a packet to travel across the network from one end to the other end, consisting of propagation and queuing delays. The load imbalance problem causes a large end-to-end delay and a large difference in delay among multiple paths. The large difference in delay brings about a significant variation in packet delay and a high risk of packet reordering (in packet-based models), leading to a large extra time introduced by the packet reordering recovery process. The packet reordering itself, large packet delay, and large variation in packet delay can significantly degrade QoS required for multimedia data transmission as well as real-time applications [29], [39], [40]. Unless otherwise stated, the term “packet delay” refers to the total packet-delay consisting of the end-to-end delay time and packet reordering recovery time, whereas “packet delay variation” refers to the variation in the end-to-end delay of packets successively arrived at a destination.

The rest of the article is organized as follows: Section 2 briefly describes existing load distribution models. Section 3 presents a new approach called Effective Delay Controlled Load Distribution (E-DCLD), enhanced from our previous work [41]. Performance of our proposed model will be compared to that of the existing models by analysis and simulations. Section 4 provides the comparative analysis. Section 5 discusses the performance evaluation under real traffic conditions. Concluding remarks are then given in Section 6.

2 Related Works

In this section, we briefly describe various load distribution models, each of which exhibits different characteristics and specific advantages (depending upon control objectives), and drawbacks. Sections 2.1 to 2.4 cover existing models, and Section 2.5 describes our previous work which is a theoretical load balancing model that will be developed into the proposed effective load distribution model.

2.1 Round Robin-Based Schemes

Surplus Round Robin (SRR) [5] is adopted from Deficit Round Robin (DRR) [42] which is a modified model from Weighted Round Robin (WRR) [15]. In SRR, a byte-based deficit counter representing the difference between the desired and actual loads (in bytes) allocated to each path is taken into account in the path selection. At the beginning of each round, the deficit counter is increased by the number of credits (referred to as quantum [5]) assigned for that path. Each time a path is selected for sending a packet, its deficit counter is decreased by the packet size. As long as the deficit counter is positive, the selection result will remain unchanged. Otherwise, the next path with the positive deficit counter will be selected in a round robin manner. If the deficit counters of all paths are nonpositive, the round is over, and a new round is started. These round robin schemes achieve starvation free (i.e., no non-work-conserving idle time) and competent load balancing efficiency; however, the major drawback is their inability to maintain per-flow packet ordering.

2.2 Least-Loaded-Based Schemes

Least-Loaded-First (LLF) [11], [12], [13] is one of the most well known load-sharing approaches introduced to handle task loads with heavy-tailed distribution, where a task is assigned to the least-loaded server. In load distribution over multiple paths, with this scheme, a path having the smallest load or the shortest queue will be selected for an arrived packet. Its major drawback is that it does not consider the order of tasks (i.e., do not keep packet ordering) as described in [14], which can result in the packet reordering problem.

2.3 Flow-Based Schemes

Direct Hashing (DH), Table-based Hashing (TH) [2], [3], [4], and Fast Switching (FS) [6] are examples of well-known flow-based models, which are simple and can completely prevent packet reordering. DH and TH are hash-based models by using hashed results of packet identifiers in a path selection. The packet identifier is obtained from the packet header information, which is typically the destination address. DH is a conventional flow-based model widely deployed in multipath routing protocols [2], [3], [4]. TH developed from DH allows us to distribute traffic in a predefined ratio by modifying the allocation of flows to paths [27]. The major drawback of these flow-based models is the inability to deal with variation of flow size distribution [8], thus leading to the load imbalance problem. In addition, the skewed distribution of destination addresses induces the load imbalance problem. FS is a table-based model which selects paths according to information in the flow-path mapping table. A packet belonging to an existing flow is sent via the same path as its preceding one. When a new flow emerges, a packet belonging to the new flow will be sent via the next parallel path in a round robin manner. Similar to DH and TH, FS can cause load imbalance due to its inability to deal with variation of the flow size distribution. However, its performance is not affected by the skewed distribution of destination addresses since it does not permanently pin a flow to a particular path by the hashed result.

2.4 Flow-Based Schemes with Adaptive Load Balancing/Distribution

Examples of adaptive load distribution models include Load Distribution over Multipath (LDM) [7], Load Balancing for Parallel Forwarding (LBPF) [8], and Flowlet Aware Routing Engine (FLARE) [9].

LDM [7], relying on [43], designed for Multiprotocol Label Switching (MPLS) networks [44] having multiple paths, randomly selects one of the multiple paths according to path utilization and hop count. A lower utilized and smaller hop-count path has a higher probability to be selected. If each flow is one packet long, performance achieved by LDM will be similar to that achieved by LLF. However, in practice, each flow is typically larger than one packet and has a different size, thus causing load imbalance among paths.

LBPF [8], in the ordinary mode, selects the path for a flow according to the hashed result of the packet identifier, similar to DH. In addition, LBPF takes into account of the traffic rate of each flow. The high-rate flows classified into a group of aggressive flows will be switched to a new path with the shortest queue at the moment when the system is under some specific condition, e.g., the system is unbalanced. Its key parameters are the size of the table which records aggressive flows, length of observation window (W), and period of adaptation (P). Load imbalance can be mitigated by setting smaller values for W and P, at the expense of packet reordering.

FLARE splits a flow into several subflows, each of which is referred to as a flowlet [9]. An interarrival time threshold calculated from a predetermined parameter (δ) and periodically measured round-trip-delay of each path (typically using ping-like operation) is used in the conditional splitting of flows; this is a key property of FLARE. A packet arrived within duration less than the interarrival time threshold is part of an existing flowlet and will be sent via the same path as the previous one. Otherwise, the packet becomes the head of a new flowlet, and is assigned to the path with the largest amount of deficit load [10]. For a smaller threshold, traffic load can be shared according to given weights; load imbalance can be reduced, however, at the expense of packet reordering, and vice versa.

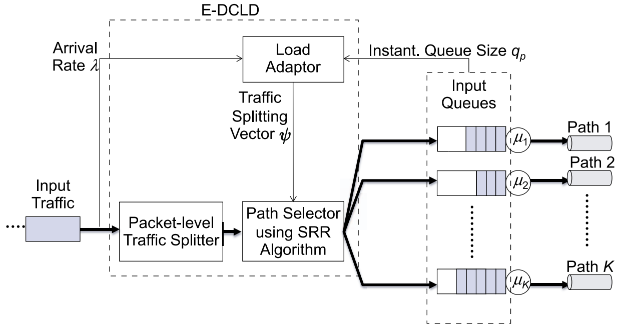

E-DCLD
Arrival Rate λ
Load Adaptor
Instant. Queue Size q_p
Traffic Splitting Vector ψ
Input Queues
Path 1
Path 2
Input Traffic
Packet-level Traffic Splitter
Path Selector using SRR Algorithm
μ_1
μ_2
μ_K
Path K

Fig. 2. Description of the proposed model, E-DCLD.

2.5 Our Previous Work

Delay Controlled Load Distribution model (DCLD) [41] uses a traffic splitting vector that determines the distribution of traffic over multiple paths, and is a theoretical idea of load balancing by calculating an optimal traffic-splitting-vector such that maximum path delay (i.e., maximum end-to-end delay) can be minimized. Unless otherwise stated, the terms, "end-to-end delay" and "path delay," are interchangeable since we assume that end-to-end delay is quasi-equal to path delay. This assumption can be held since delays experienced by two successive packets sent via the same path are likely similar, whereas delays of those sent via different paths having unequal delays are likely to be dissimilar. DCLD computes the path delay by using the M/M/1 queuing model, and reduces the difference among path delays by decreasing load assigned to the path with the largest delay and increasing load by the same amount (of the reduced load) to the other path with the smallest delay. Traffic splitting ratios are thereby gradually adjusted until all path delays are equal. However, DCLD was designed for Poisson traffic, and is thus likely not practical for a real network under different traffic conditions (e.g., non-Poisson traffic, bursty traffic, and so on).

3 Proposed Model

Since solutions to efficiently control packet delay in load distribution has not been widely studied, several problems regarding the delay such as large packet delay and large variation among packet delays are yet to be addressed. In order to provide efficient load balancing to determine the optimal traffic splitting vector, we have proposed our previous work, DCLD [41], which still has some drawbacks. In this paper, we propose E-DCLD enhanced from DCLD that can overcome the drawbacks of DCLD and outperform the existing models in solving the delay-related problems. Fig. 2 shows the functional block diagram of E-DCLD. E-DCLD takes into account of input traffic rate and the instantaneous queue size, which are locally available information, in determining the traffic splitting vector, and thereby properly responding to network condition without additional network overhead. In the path selector, we implement the SRR load sharing algorithm [5] which does not restrict weights to be integers. This is suitable for our work since the calculated traffic splitting vector is typically not an integer. The traffic splitting vector determination and adaptive load adaptation algorithms, which are improved from DCLD, are detailed as follows:

Let $\mathbf{P}$ be a set of multiple paths. For $\forall p \in \mathbf{P}$, we formulate the cost function of path p, which is a function of the estimated end-to-end delay consisting of the fixed delay and the variable delay,

$$C_p(\psi_p) = D_p + (1-w)\frac{1}{\mu_p - \psi_p\lambda} + w\frac{q_p}{\mu_p}. \tag{1}$$

The fixed delay (i.e., propagation delay) of path p is the first term, denoted by D_p. The variable delay focused in our work is the queuing delay which varies according to the input traffic rate (λ), the bandwidth capacity of the path (μ_p), and the traffic splitting ratio (ψ_p). With the assumption that input traffic is a combination of Poisson traffic and unknown traffic which cannot be identified, the queuing delay is modeled as a mixture of an M/M/1 queue (which has low complexity as compared to other queuing models) and a measurement. Therefore, with a weight factor w, the queuing delay is obtained by averaging the second term which is the average queuing delay derived from the M/M/1 model and the third term which is the waiting time of the current packet at an input queue having queue size of q_p with unknown queuing model, thus measured as q_p/μ_p. With a small value, $w \rightarrow 0$, E-DCLD calculates the queuing delay by using the M/M/1 model, which is similar to the DCLD model and is accurate under the Poisson traffic condition. On the other hand, with a large value, $w \rightarrow 1$, the queuing delay is calculated only from the queue size, which is almost similar to the LLF model that can decrease the average queue size but is likely to increase the risk of packet reordering.

From (1), the optimal splitting vector can be derived by solving the optimization problem as follows:

$$\begin{aligned} &\text{Minimize} && \max_{p\in\mathbf{P}} C_p(\psi_p), \\ &\text{subject to} && \sum_{p\in\mathbf{P}} \psi_p = 1, \\ &\text{and} && 0 \le \psi_p \le \frac{\mu_p}{\lambda} \le 1. \end{aligned} \tag{2}$$

The traffic splitting vector, $\boldsymbol{\psi}^n = \{\psi_p^n\}$ for all $p \in \mathbf{P}$, consists of the control variables of the problem described in (2) and the proportion of traffic allocated to path p at time t_n. The initial splitting vector, $\boldsymbol{\psi}^0$, is calculated from (3)

$$\forall p \in \text{P} : \psi_p^0 = \frac{\mu_p}{\sum_{p\in\mathbf{P}} \mu_p}. \tag{3}$$

When the mth packet arrives (at a diverging point of input traffic), the packet arrival rate λ and instantaneous queue size q_p measured from the input traffic and the input queue, respectively, are used to calculate the estimated end-to-end delay of each path according to (1). While the traffic load is distributed to the multiple paths in a round robin manner, the load adaptor decreases load on the path having the largest estimated delay (i.e., p_{worst}), and then increases load on the path having the smallest estimated delay (i.e., p_{best}) by the same amount of the reduced load. Change of path costs can be illustrated in Fig. 3a. For each arrived packet, the load adaptor performs the load adaptation algorithm (to adjust the traffic splitting vector) which can be described in the following steps:

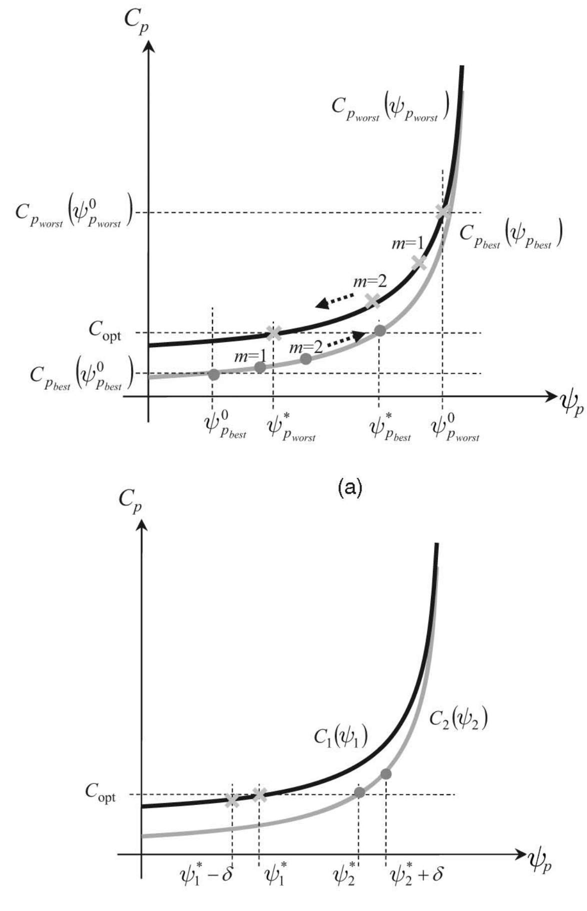


Fig. 3. Change of path costs. (a) From the beginning to the equilibrium point. (b) Away from the equilibrium point.

1. Calculate $C_p(\psi_p)$ by using (1) for each $p \in \mathbf{P}$.
2. Among all paths,
   select $p_{worst} \in \mathbf{P}$ having the maximum cost and
   select $p_{best} \in \mathbf{P}$ having the minimum cost.
3. Calculate $\Delta\psi$ such that

   $$C_{p_{worst}}(\psi_{p_{worst}} - \Delta\psi) = C_{p_{best}}(\psi_{p_{best}} + \Delta\psi), \tag{4}$$

   The solution, $\Delta\psi$, is presented in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.43.
4. To avoid a negative value of the traffic splitting ratio on path p_{worst} (i.e., $\psi_{p_{worst}} < 0$) and overload on path p_{best} (i.e., $\psi_{p_{best}} > \mu_{p_{best}}/\lambda$), $\Delta\psi$ must be appropriately determined by

   $$\Delta\psi \leftarrow \min(\psi_{p_{worst}}, \Delta\psi), \quad \text{and then}$$
   $$\Delta\psi \leftarrow \min\Big(\frac{\mu_{p_{best}}}{\lambda} - \psi_{p_{best}}, \Delta\psi\Big).$$

5. Update $\psi^m_{p_{worst}} = \psi^{m-1}_{p_{worst}} - \Delta\psi$ and $\psi^m_{p_{best}} = \psi^{m-1}_{p_{best}} + \Delta\psi$.
   For all paths $p \in P$ except p_{best} and $p_{worst}, \psi_p^m = \psi_p^{m-1}$.

When $m \rightarrow \infty$, the cost of each path will converge to the same value, which allows us to achieve the objective function in (2). The proof of convergence of E-DCLD is presented in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.43. Next, we will show that the equilibrium of the load adaptation algorithm is equal to the optimum. This can be explained by proving that, from the equilibrium point, further decrease of ψ_p will cause the largest cost (among all paths) to increase from the minimum value.

Proof. Assume that there are two paths, i.e., $\mathbf{P} = \{1, 2\}$, having cost functions illustrated in Fig. 3b and $C_1(\psi_1^0) > C_2(\psi_2^0)$. At the equilibrium point, with optimal traffic splitting vector ψ^*, we obtain $C_1(\psi_1^*) = C_2(\psi_2^*) = C_{\text{opt}}$ and

$$\max_{p \in \mathbf{P}} C_p(\psi_p^*) = C_{\text{opt}}.$$

When we further transfer load, δ, from path 1 to path 2, i.e., $\psi_1 = \psi_1^* - \delta$ and $\psi_2 = \psi_2^* + \delta$, since $C_p(\psi_p)$ is a monotonically increasing function of $\psi_p, C_2(\psi_2) = C_2(\psi_2^* + \delta) > C_2(\psi_2^*) = C_{\text{opt}}$. Therefore,

$$\max_{p \in \mathbf{P}} C_p(\psi_p) > C_{\text{opt}}.$$

□

This proof is also valid when there are more than two paths. Some numerical results of DCLD which is a simplified version of E-DCLD are presented in [41].

4 Analysis

In this section, we analyze the performance of E-DCLD and present simulation-based verifications, in terms of end-to-end delay, packet delay variation, risk of packet reordering, and total packet delay. First, we show that E-DCLD can reduce end-to-end delay. Then, we show that it can also reduce variation in end-to-end delay, which allows us to achieve smaller variation in packet delay and relatively low risk of packet reordering among packet-based models.

To verify the analysis, we conduct simulations under the environment as shown in Fig. 1 from the view point of a source having multiple paths to a destination. The input traffic from the source will be split into three multiple paths ($K = 3$) having aggregated bandwidth (μ) of 8 Mbps and having ratios of bandwidth capacity (among the parallel paths) of 1:2:3. The service time of a packet is assumed to be exponentially distributed where the mean service time is inversely proportional to the bandwidth capacity, i.e., $1/\mu$. With the multiple paths, each load distribution model is 1-hour-long simulated under the load condition varying from low to high. Input traffic consists of three independent Poisson flows, each of which has the ratio of mean packet arrival rate corresponding to that of the bandwidth capacity of the parallel paths, i.e., 1:2:3, where the mean packet arrival rate is chosen such that the ratio of the mean offered load to the mean service rate (λ/μ) varies from 0.1 to 0.9 with a step size of 0.1 for each simulation round of each model. We assume that all paths have no fixed-delay (i.e., zero propagation delay) since its effect on determination of the traffic splitting vector has already been discussed in [41]. For all simulations, the runtime parameter for E-DCLD, w, is chosen to be 0.5, and parameters for candidate models are chosen by following the guidelines in their respective papers. SRR, LLF, FS, LBPF, and FLARE are candidates for comparisons. In SRR, the numbers of credits assigned for path 1, path 2, and path 3 are 1, 2, and 3, respectively, corresponding to bandwidth capacities of the paths. In LBPF, the size of the table for recording aggressive flows is 1, the length of the observation window (W) is 1,000, and period of adaptation (P) is 20; that is, the table will be updated for every 1,000 packets and the largest flow recorded in the table will be switched to a new path for every 20 packets.

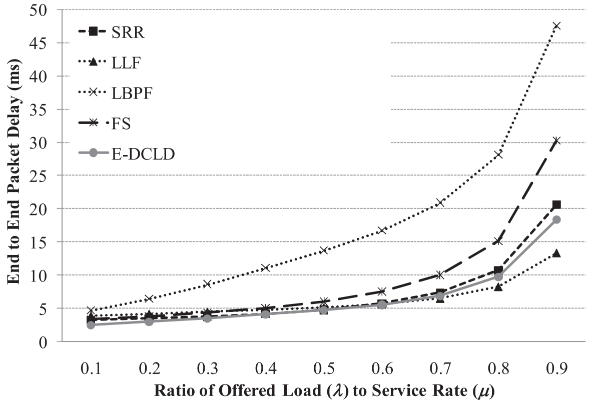


Fig. 4. Mean end-to-end delay when input traffic is Poisson.

4.1 End-to-End Delay

Let $D_p^{(m)}$ and $Q_p^{(m)}$ be propagation delay and queuing delay, respectively. They constitute the end-to-end delay $d_p^{(m)}$ (i.e., $d_p^{(m)} = D_p^{(m)} + Q_p^{(m)}$) that is experienced by the mth packet sent via path p; d_p is the expected value of the path delay averaged over m packets. Theoretically, if the input traffic is Poisson and path p is randomly selected with probability ψ_p while at least one packet is being forwarded via the path, with the assumption that $1/\mu_p$ is the (expected) service time in sending a packet to its destination and q_p/μ_p is the (expected) waiting time of the packet in the queue, the cost value obtained from the cost function C_p in (1) will be close to the (expected) end-to-end delay of path p, i.e., d_p. In a long-run system where the rate of input traffic is quasi-static during a short update-period, with the optimal traffic splitting vector ψ^*, all paths have (almost) the same delay. The maximum path delay is minimized and the end-to-end delay is therefore reduced.

Fig. 4 compares the means of end-to-end delays achieved by various models. E-DCLD achieves smaller end-to-end delay than that of SRR even though weights (i.e., quantum [5]) chosen in SRR are proportional to bandwidth capacities of the multiple paths. Among the packet-based models, LLF is possible to keep a small end-to-end delay since only the path having the smallest queue size is selected for sending a packet. LLF selects the path based on the queue size and should be able to maintain the smallest end-to-end delay. Only under the condition of high load, LLF achieves a little bit smaller delay than that of E-DCLD. Fig. 4 also shows that flow-based models like FS and LBPF incur large delay due to variation in the flow size distribution. The simulation environment of FS is set up such that FS achieves near-perfect load balance; however, its end-to-end delay is still large. Note that the simulated environment of FS is not

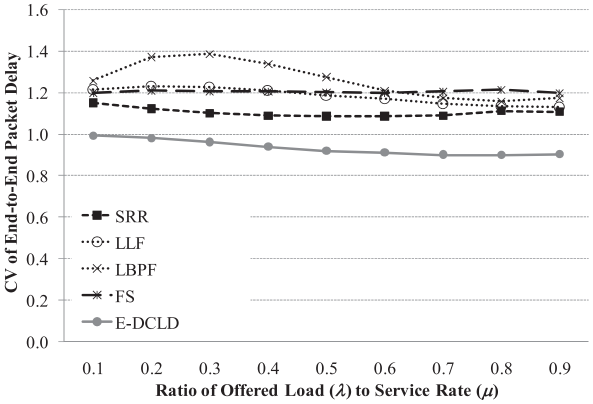


Fig. 5. Coefficient of variation of end-to-end delay when input traffic is Poisson.

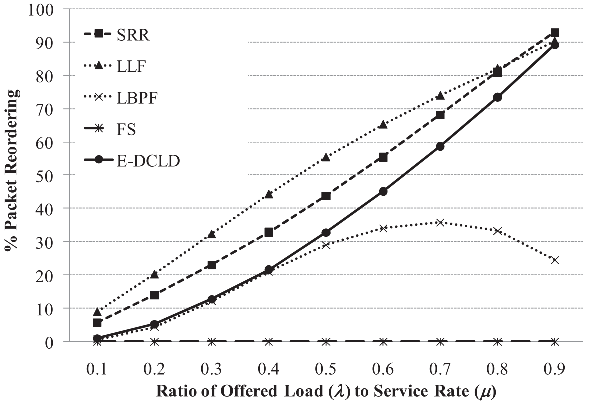


Fig. 6. Risk of packet reordering when input traffic is Poisson.

compatible with a real network, implying that its end-to-end delay is likely to be much larger than that in the simulation.

4.2 Packet Delay Variation

Here, let $\Delta_{i,j}$ be the expected value of $\Delta_{i,j}^{(m)}$, i.e., $\Delta_{i,j}^{(m)} = d_i^{(m-1)} - d_j^{(m)}$ for $\forall j \neq i$. Since E-DCLD tries to minimize the difference among path delays of all paths, $|\Delta_{i,j}|$ is thus reduced. As compared to E-DCLD as well as the other packet-based models, flow-based models can cause large variation in packet delay, affected from overload and, consequently, large end-to-end delay on a particular path. Fig. 5 presents the coefficient of variation (CV) among end-to-end delays of all candidates. E-DCLD aiming to reduce $|\Delta_{i,j}|$ achieves the least delay variation. On the other hand, SRR, LLF, FS, and LBPF having larger $|\Delta_{i,j}|$ are likely to cause larger variation. In LBPF, taking queue sizes into account in load balancing, when λ/μ is so small that all queues are empty, traffics (each with a different rate) are carried by the same path, thus incurring large variation. When λ/μ increases such that all queues are occupied, traffics are distributed; the variation is thus decreased. LLF uses the similar path selection scheme, and hence the same trend of variation is observed; however, since LLF is packet-based, the degree of variation is smaller as compared to that of LBPF.

4.3 Risk of Packet Reordering

Risk of packet reordering affects the number of reordered packets as well as the degree of packet reordering, and thus incurs packet reordering recovery time. In this section, risk of packet reordering will be analyzed. Effect of packet reordering recovery time on the total packet delay will be described in the next section.

Derived in [38], the risk of packet reordering can be presented in terms of the probability of packet reordering, π_{r}, as follows:

$$\pi_{\mathrm{r}} = \pi_s \sum_{i \in \mathbf{P}} \sum_{j \in \mathbf{P}} \Phi_{i,j}^{(\mathrm{m})} \Omega\big(\Delta_{i,j}^{(m)}\big), \tag{5}$$

where π_{s} is the probability of splitting and $\Phi_{i,j}^{(m)}$ is the probability of the path switching from path i to path j (i.e., paths i and j are selected for the $(m-1)$th packet and the mth packet, respectively), depending on the path selection strategy; $\Omega(\Delta_{i,j}^{(m)})$ denotes the conditional probability of packet reordering when the path is switched from path i to path j, and is a function of $\Delta_{i,j}^{(m)}$, i.e., the difference of end-to-end delays between path i and path j. As described in [38], $\Omega(\Delta_{i,j}^{(m)})$ is the cumulative distribution function of the packet interarrival time; if $\Delta_{i,j}^{(m)} > 0, \Omega(\Delta_{i,j}^{(m)}) > 0$ implies that there is a risk of packet reordering; otherwise, $\Omega(\Delta_{i,j}^{(m)}) = 0$, that is, packet reordering will never occur. The smaller value of $\Delta_{i,j}^{(m)}$, the smaller risk of packet reordering; therefore, E-DCLD aiming to minimize $\Delta_{i,j}$ strives to maintain a low risk of packet reordering. As compared to E-DCLD, packet-based models such as SRR and LLF can cause a high risk of packet reordering. Especially, LLF, which only chooses the path with the shortest queue, is highly likely to have $\Delta_{i,j}^{(m)} > 0$, implying that it can cause a high risk of packet reordering.

Fig. 6 shows that E-DCLD, which can decrease the variation among end-to-end delays as illustrated in Fig. 5, can thus reduce the risk of packet reordering while the other packet-based models like SRR and LLF incurring large variation among end-to-end delays induce a high risk of packet reordering. The variation in the end-to-end delay does not induce risk of packet reordering for FS which does not change path for all packets in the same flow, but does induce the risk of packet reordering for LBPF which allows a flow to be split. In LBPF, when λ/μ increases, π_{s} increases; on the other hand, the probability of having idle period on each path decreases, thus reducing the probability of path change, i.e., $\Phi_{i,j}^{(m)}$ decreases while $\Phi_{i,j}^{(m)}$ increases. When λ/μ is large, further increase of λ/μ can cause $\Phi_{i,j}^{(m)}$ to decrease significantly, thus reducing the rate of increase of π_r.

4.4 Total Packet Delay

The total packet delay is the delay experienced by users. It includes two factors: end-to-end delay and additional time delay required for packet ordering recovery. E-DCLD aims to decrease both of the two factors and can thus efficiently reduce the total packet delay. SRR and LLF can cause a high risk of packet reordering, and consequently require long time for packet reordering recovery, whereas FS, LBPF, and FLARE can cause a large end-to-end delay. As illustrated in Fig. 7, E-DCLD achieves both low end-to-end delay and low risk of packet reordering, and thus can maintain a small (total) packet delay.

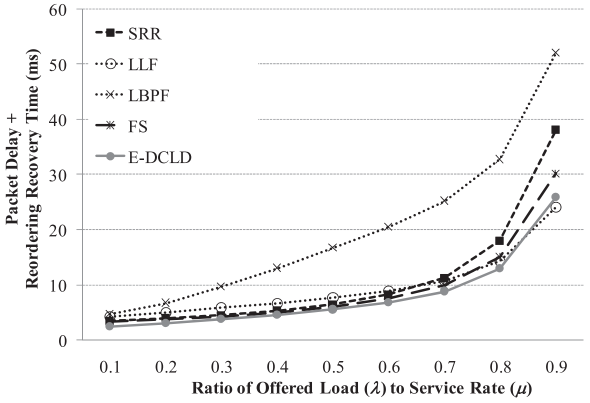


Fig. 7. Mean total (packet) delay when input traffic is Poisson.

5 Real-Traffic-Based Performance Evaluation

In this section, comparative performance under various conditions of real traffics which are not Poisson is demonstrated and discussed. Simulation setup in this section is almost similar to that in the previous section with the following exceptions. Five simulation scenarios are conducted to show the performance of each load distribution model, by using 1-hour long real traffic traces [45], i.e., DS1, DS2, DS3, DS4, and DS5, which contain wide-area traffics at primary Internet access point between Digital Equipment Corporation and the rest of the world, where characteristics of the traces are listed in Table 1. Bandwidth capacities (or mean service rates) of path 1, path 2, and path 3 are 1, 4, and 7 Mbps, respectively; the total bandwidth capacity of the multiple paths is 12 Mbps. As compared to the bandwidth capacities, traffics generated from trace DS1 and DS2 cause moderate load whereas those generated from trace DS3 and DS4 incur heavy load and some load-spikes. Moreover, we use trace DS5 to generate extremely heavy traffic, having maximum offered load much higher than the total bandwidth capacity, thus incurring overload on the multiple paths.

With the setup simulation environment, E-DCLD, SRR, LLF, LBPF, and FLARE are evaluated. In SRR, the numbers of credits assigned for path 1, path 2, and path 3 are 1, 4, and 7, respectively. In LBPF, the size of the table is 20, $W = 1{,}000$, and $P = 20$. In FLARE, δ is set to 50 ms (i.e., minimum of interarrival time threshold), the numbers of credits assigned for the paths are similar to those in SRR, and round-trip-delay is examined every 500 ms. Since performance of LBPF and FLARE is better than that of a conventional flow-based model, LBPF and FLARE will be used as representatives of flow-based models in the comparisons. Simulations in Section 5.1 are conducted to evaluate E-DCLD with equal fixed delays (which are assumed to be 0 for simplicity) in order to specifically emphasize the advantage of the additional component of E-DCLD over DCLD, whereas those with different fixed delays in Section 5.2 are conducted to demonstrate the superior performance of E-DCLD in such a realistic environment.

TABLE 1
Profile of Traffic Traces [45]

Trace ID	# Packets x10^6	Traffic Rate (Mbps.)			# Different Flows	Flow Size (Packets)		Flow Rate (Flows/Second)		
		Mean	Min.	Max.		Mean	CV	Mean	Min.	Max.
DS1	0.83	1.84	0.82	3.58	38032	21.82	16.13	145.23	77	209
DS2	1.19	2.64	0.55	3.68	58025	20.46	33.09	174.85	50	257
DS3	2.66	5.91	2.07	13.65	5865	453.87	7.52	137.89	77	204
DS4	2.87	6.38	0.46	12.24	12903	222.71	5.98	175.32	44	247
DS5	3.86	8.58	1.86	15.45	12710	303.88	7.11	184.50	90	269

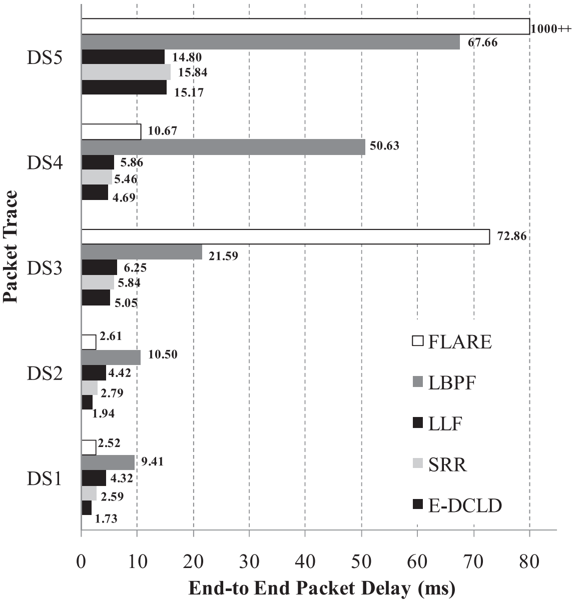


Fig. 8. Mean end-to-end delay under input traffic generated from traces of real traffic ($D_1 = D_2 = D_3 = 0$).

5.1 Equal Fixed Delays

In this simulation, all fixed delays are assumed to be equal: $D_1 = D_2 = D_3 = 0$.

5.1.1 End-to-End Delay

Fig. 8 shows that E-DCLD achieves smaller end-to-end delay as compared to the other models. LBPF and FLARE, which are flow-based models, cause congestion and thus lead to a large delay even though they try to split large flows and dynamically adjust the amount of load assigned on each path. As compared to LBPF, FLARE decreases the probability of splitting dramatically as the input traffic rate increases significantly with input traffics generated from traces DS3 and DS5, which have large mean and variation of flow size distribution.

Among packet-based models, LLF, which selects the path with the smallest queue size, should achieve the smallest delay. However, in practice, the instantaneous queue size does not always accurately reflect the path delay; in other words, time taken for sending a packet via a path having the smallest queue size is not always minimal. As compared to E-DCLD, LLF has comparable performance only if the network is so congested that all paths have long queues as shown by the simulation results under the condition of heavy traffic generated from trace DS5. However, in most cases, E-DCLD taking into account of

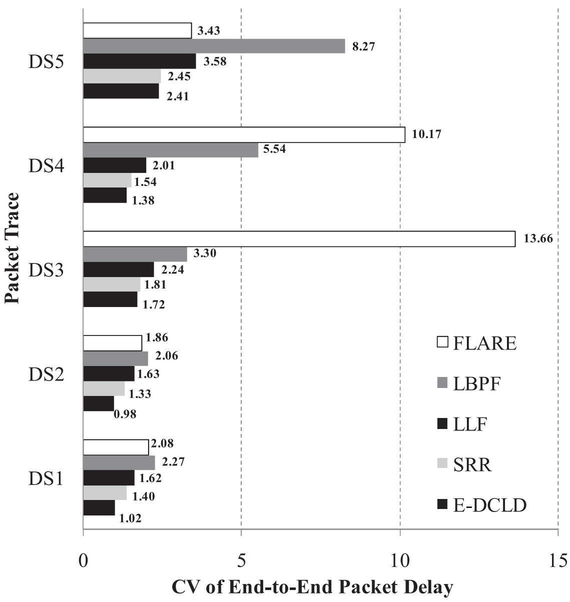


Fig. 9. Coefficient of variation of end-to-end delay under input traffic generated from traces of real traffic $(D_1 = D_2 = D_3 = 0)$.

input traffic and queue size in calculating path delay can decrease the end-to-end delay. As compared to SRR, E-DCLD with adaptive weight adjustment using our proposed load adaptation algorithm can decrease the end-to-end delay.

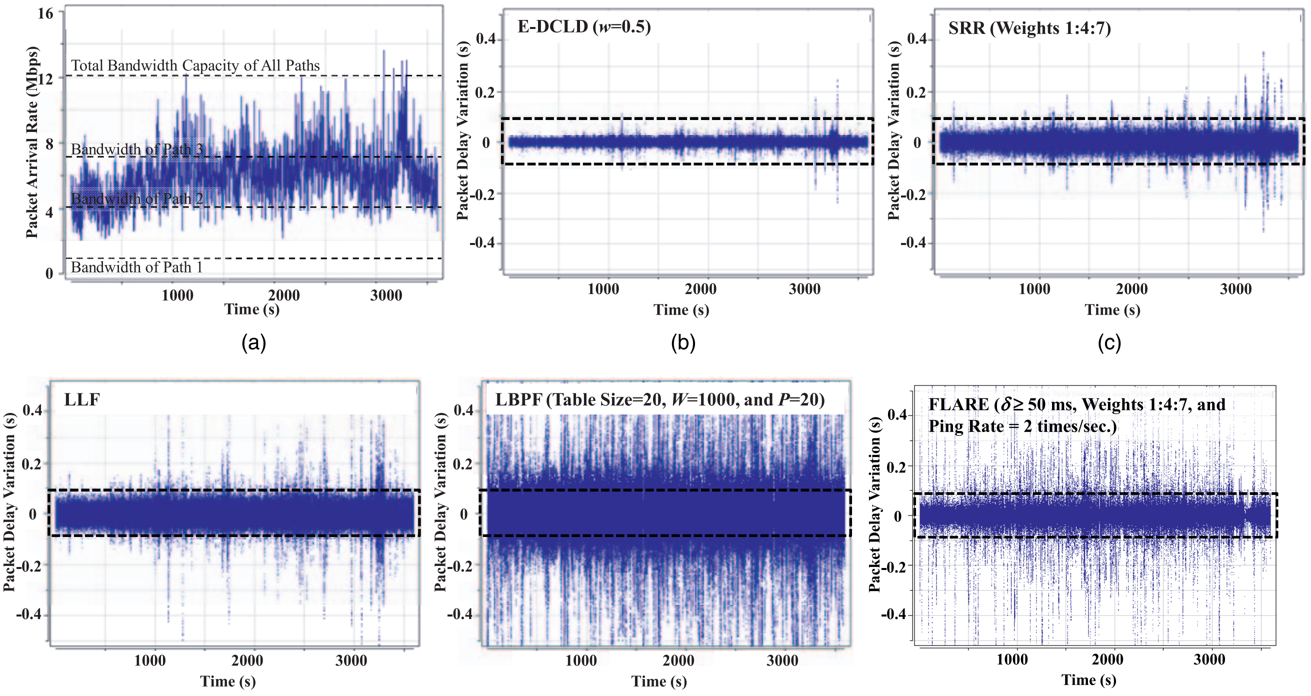
5.1.2 Packet Delay Variation

Fig. 9 shows that E-DCLD maintains low variation among end-to-end delays as compared to the variations caused by the other candidates. In the LLF model, choosing only the path with the smallest queue still causes larger variation of the end-to-end delay. In LBPF and FLARE, congestion or overload on a particular path causes a significantly large degree of variation, especially, under heavy load induced by traffic traces DS3, DS4, and DS5. Moreover, Fig. 10 shows that E-DCLD can efficiently mitigate variation in the end-to-end delay caused by the overloaded paths. Fig. 10a illustrates the raw traffic generated from trace DS3 as well as the capacities of path 1, 2, and 3, and the total capacity of multiple paths. Figs. 10b, 10c, 10e, and 10f demonstrate the performance among all models, and the evidence that E-DCLD can maintain the smallest delay variation. Under various traffic conditions, Fig. 11 shows packet delay variations achieved by various models, and thus clearly demonstrates the superiority of E-DCLD.

5.1.3 Risk of Packet Reordering

Fig. 12 illustrates that E-DCLD can efficiently alleviate packet reordering which inherently exists in packet-based models such as SRR and LLF. SRR, which sends packets in a round robin manner, does not have any additional mechanism to prevent packet reordering, and consequently causes a high risk of packet reordering. LLF, which chooses only the path with the shortest queue size, also causes a very high risk of packet reordering.

Theoretically, flow-based models which send all packets belonging to the same flow via the same path have no risk of packet reordering. However, variants of flow-based models allow switching a path for some of the packets to improve load balancing efficiency at the price of a risk of packet reordering. The trade-off between improving load balancing and maintaining a low risk of packet reordering depends on the respective algorithms as well as their set



(a) (b) (c)

(d) (e) (f)

Fig. 10. (a) Characteristic of traffic generated from traffic trace DS3 available online [45]. (b)-(f) Packet delay variation under traffic generated from trace DS3 when load distribution models, E-DCLD, SRR, LLF, LBPF, and FLARE, are employed, respectively, $(D_1 = D_2 = D_3 = 0)$.

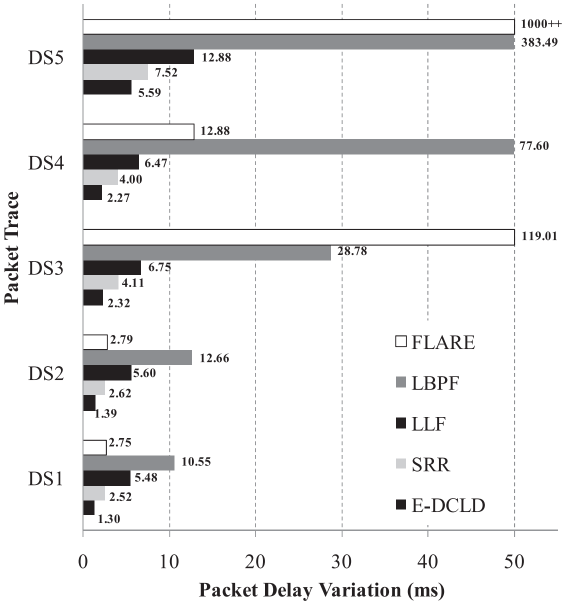


Fig. 11. Packet delay variation under input traffic generated from traces of real traffic $(D_1 = D_2 = D_3 = 0)$.

parameters. LBPF splits a group of largest flows, thus causing the risk of packet reordering. FLARE splits only flows with packet interarrival time which is small enough, and hence does not cause packet reordering [6], [8], thus minimizing the risk of packet reordering.

5.1.4 Total Packet Delay

Similar to the results of simulations conducted under the condition of Poisson traffic, the total (packet) delay achieved by various models is illustrated in Fig. 13. E-DCLD, having both low end-to-end delay and low risk of packet reordering, exhibits superiority in mitigating the total packet delay as compared to the other models. The other packet-based models (such as SRR and LLF) have a high risk of packet reordering, thus leading to a large total delay whereas flow-based models (such as LBPF and FLARE) incur a large total delay because of a large end-to-end delay and a large degree of variation in the end-to-end delay.

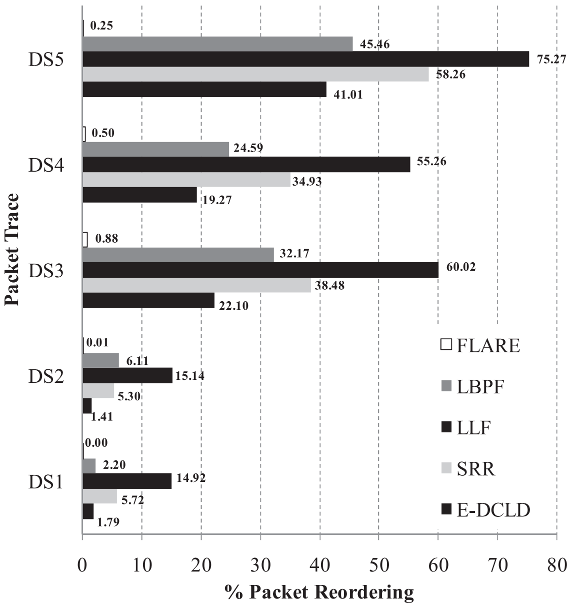


Fig. 12. Risk of packet reordering under input traffic generated from traces of real traffic $(D_1 = D_2 = D_3 = 0)$.

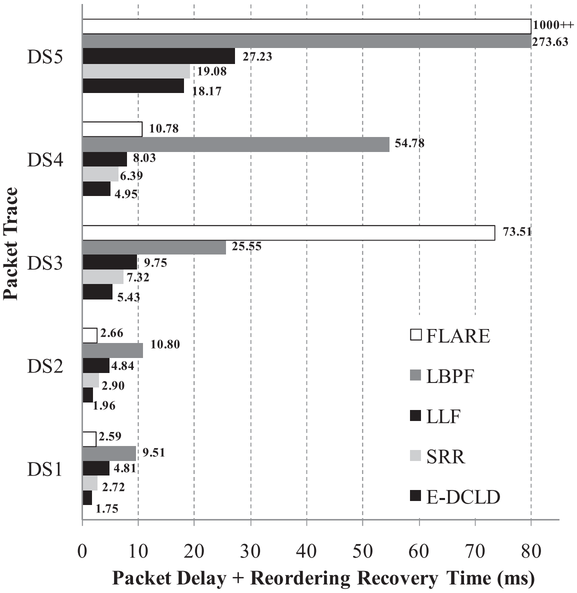


Fig. 13. Mean total (packet) delay under input traffic generated from traces of real traffic $(D_1 = D_2 = D_3 = 0)$.

5.2 Unequal Fixed Delays

In this simulation, each path is assumed to have different fixed delays: $\mathrm{D}_1 = 1$ ms, $\mathrm{D}_2 = 2$ ms, and $\mathrm{D}_3 = 3$ ms; path 1 has the smallest bandwidth but has the smallest fixed delay whereas path 3 has the largest bandwidth but has the largest fixed delay. The fixed delay becomes one of the key parameters in determining the traffic splitting vectors in the E-DCLD model. Table 2 shows that the number of packets sent via path 3 decreases while the numbers of packets sent via path 1 and path 2 increase, as compared to the results when all fixed delays are equal. This indicates the change of preference for the paths. Next, we examine E-DCLD's performance; the results show that E-DCLD still outperforms the other models. E-DCLD can reduce the end-to-end delay (as illustrated in Fig. 14) and variation among the

TABLE 2
Simulation Results of E-DCLD: Ratio of the Number of Packets Sent via Each Path when Fixed Delays Are Different

Trace ID	Fixed Delays: D_1=D_2=D_3=0			Fixed Delays: D_1=1ms, D_2=2ms, D_3=3ms		
	# Packets Sent via Path 1 (%)	# Packets Sent via Path 2 (%)	# Packets Sent via Path 3 (%)	# Packets Sent via Path 1 (%)	# Packets Sent via Path 2 (%)	# Packets Sent via Path 3 (%)
DS1	0.00	6.76	93.24	0.00	32.17	67.82
DS2	0.00	9.45	90.55	0.00	33.64	66.36
DS3	0.93	28.32	70.75	1.18	35.38	63.44
DS4	0.87	29.49	69.64	1.16	34.81	64.03
DS5	3.45	32.48	64.06	3.93	33.55	62.52

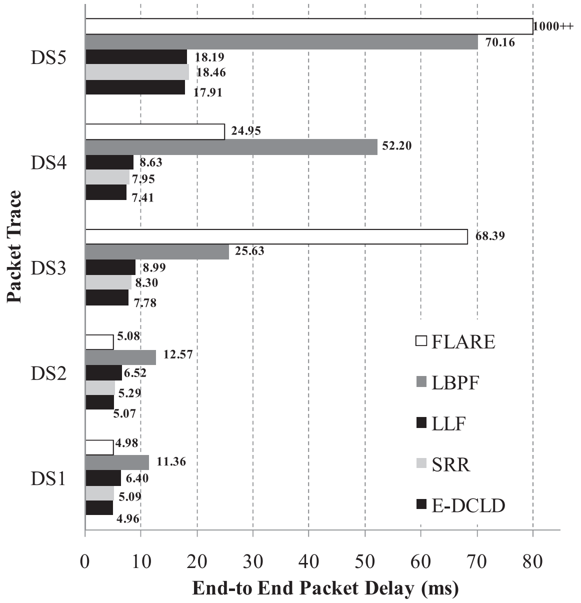


Fig. 14. Mean end-to-end delay under input traffic generated from traces of real traffic ($D_1 = 1$ ms, $D_2 = 2$ ms, and $D_3 = 3$ ms).

end-to-end delays (as illustrated in Fig. 15) such that the packet delay variation and risk of packet reordering can be significantly reduced, as illustrated in Figs. 16 and 17, respectively. Likewise, the packet delay can be decreased as illustrated in Fig. 18. As observed in Figs. 14 and 15, while E-DCLD and FLARE have the same mean end-to-end delay, E-DCLD exhibits a much smaller variation in the end-to-end delay; this observation differentiates their performances in long and short time scales. Although FLARE, similar to E-DCLD, can maintain a small end-to-end delay in long time scale, it can cause a large delay in short time scale. This is attributed to their different traffic splitting and path selection schemes.

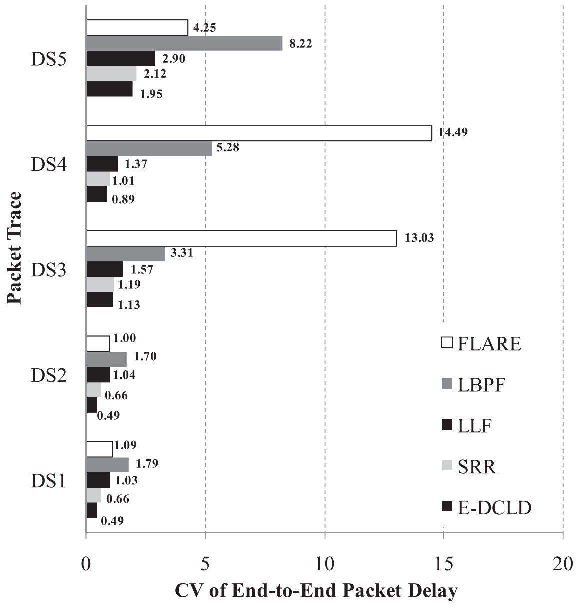


Fig. 15. Coefficient of variation of end-to-end delay under input traffic generated from traces of real traffic ($D_1 = 1$ ms, $D_2 = 2$ ms, and $D_3 = 3$ ms).

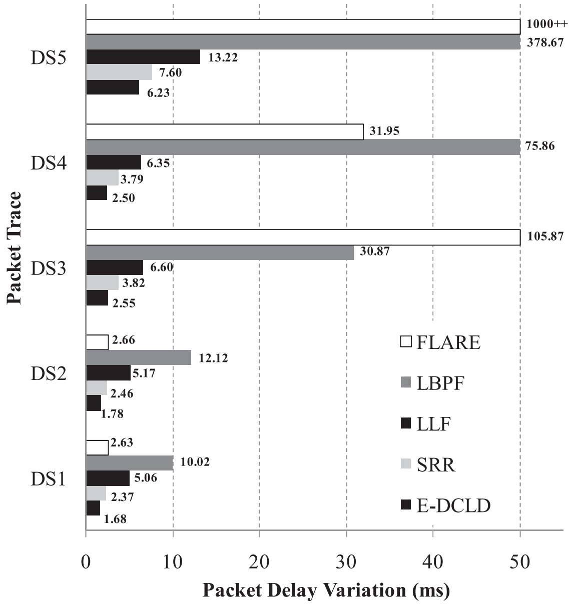


Fig. 16. Packet delay variation under input traffic generated from traces of real traffic ($D_1 = 1$ ms, $D_2 = 2$ ms, and $D_3 = 3$ ms).

6 Concluding Remarks

Since an effective model of load distribution is critical to efficiently utilize multiple available paths for multimedia data transmission and real-time applications which are sensitive to packet delay, packet delay variation, and packet reordering, we have proposed a novel load distribution model, E-DCLD, which aims to minimize the difference among end-to-end delays by using locally available information. By doing so, the packet delay variation can be reduced and thus the risk of packet reordering is minimized, without incurring additional network overhead. When the risk of packet reordering is small, the extra time required for the packet reordering recovery process is likely small. Therefore,

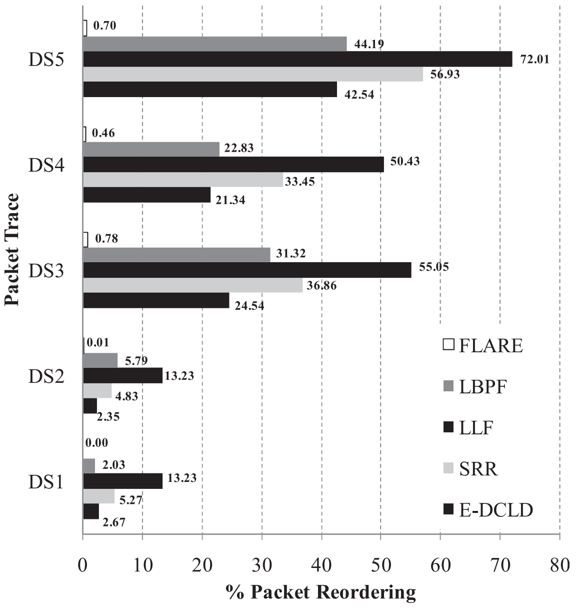


Fig. 17. Risk of packet reordering under input traffic generated from traces of real traffic ($D_1 = 1$ ms, $D_2 = 2$ ms, and $D_3 = 3$ ms).

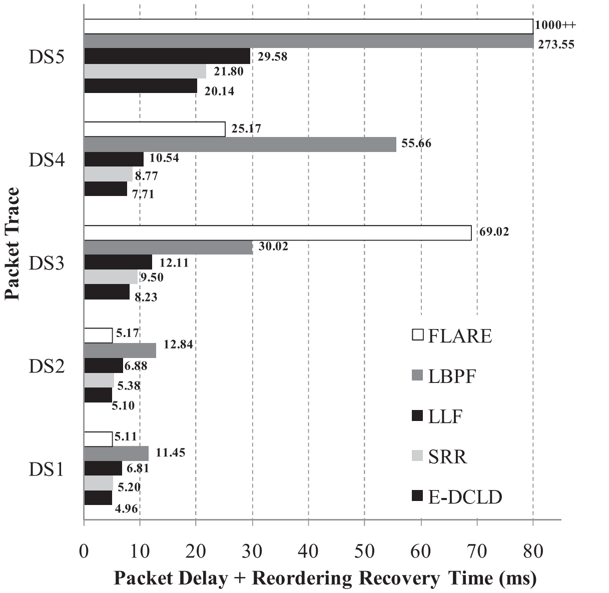


Fig. 18. Mean total (packet) delay under input traffic generated from traces of real traffic ($D_1 = 1 \text{ ms}, D_2 = 2 \text{ ms}, \text{ and } D_3 = 3 \text{ ms}$).

minimizing the difference of end-to-end delays can maintain not only a small end-to-end delay but also the packet reordering recovery time. In order to justify the superior performance of E-DCLD, we have provided comparative performance among E-DCLD and the current existing models by analysis and by simulations under various traffic conditions. For the future work, since E-DCLD does not contain any complex component, it can be incorporated into various applications, e.g., load balancing in multipath transport protocols, with low implementation complexity.

REFERENCES

[1] L. Golubchik, J. Lui, T. Tung, A. Chow, W. Lee, G. Franceschinis, and C. Anglano, "Multi-Path Continuous Media Streaming: What Are the Benefits?" *Performance Evaluation,* vol. 49, pp. 429-449, Sept. 2002.

[2] C. Villamizar, "OSPF Optimized Multipath (OSPF-OMP)," Internet Draft Draft-Ietf-Ospf-Omp-02.txt, vol. ospt, no. 2, pp. 1-38, Feb. 1999.

[3] D. Thaler and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection," RFC 2991, Nov. 2000.

[4] C. Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992, Nov. 2000.

[5] H. Adiseshu, G. Parulkar, and G. Varghese, "A Reliable and Scalable Striping Protocol," *ACM SIGCOMM Computer Comm. Rev.,* vol. 26, no. 4, pp. 131-141, Oct. 1996.

[6] A. Zinin, *Cisco IP Routing, Packet Forwarding and Intra-Domain Routing Protocols.* Addison Wesley, 2002.

[7] J. Song, S. Kim, M. Lee, H. Lee, and T. Suda, "Adaptive Load Distribution over Multipath in MPLS Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '03),* pp. 233-237, May 2003.

[8] W. Shi, M.H. MacGregor, and P. Gburzynski, "Load Balancing for Parallel Forwarding," *IEEE/ACM Trans. Networking,* vol. 13, no. 4, pp. 790-801, Aug. 2005.

[9] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic Load Balancing without Packet Reordering," *ACM SIGCOMM Computer Comm. Rev.,* vol. 37, no. 2, pp. 53-62, Apr. 2007.

[10] K.C. Leung and V.O.K. Li, "Generalized Load Sharing for Packet-Switching Networks: Theory and Packet-Based Algorithm," *IEEE Trans. Parallel and Distributed System,* vol. 17, no. 7, pp. 694-702, July 2006.

[11] K.G. Shin and C.J. Hou, "Design and Evaluation of Effective Load Sharing in Distributed Real-Time Systems," *IEEE Trans. Parallel and Distributed Systems,* vol. 5, no. 7, pp. 704-719, July 1994.

[12] O. Kremien and J. Kramer, "Methodical Analysis of Adaptive Load Sharing Algorithms," *IEEE Trans. Parallel and Distribution Systems,* vol. 3, no. 6, pp. 747-760, Nov. 1992.

[13] C.C. Hui and S.T. Chanson, "Hydrodynamic Load Balancing," *IEEE Trans. Parallel and Distributed Systems,* vol. 10, no. 11, pp. 1118-1137, Nov. 1999.

[14] Z. Tari, J. Broberg, A. Zomaya, and R. Baldoni, "A Least Flow-Time First Load Sharing Approach for Distributed Server Farm," *J. Parallel and Distributed Computing,* vol. 65, no. 7, pp. 832-842, July 2005.

[15] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case," *IEEE/ACM Trans. Networking,* vol. 1, no. 3, pp. 344-357, June 1993.

[16] G. Malkin, "RIP Version 2," RFC 2453, Nov. 1998.

[17] J. Moy, "OSPF Version 2," RFC 2328, Apr. 1998.

[18] Cisco Systems Inc., "Enhanced Interior Gateway Routing Protocol (EIGRP),"Cisco white paper EIGRP, http://www.cisco.com/warp/public/103/eigrp-toc.html or http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094cb7.shtml, 2011.

[19] J. Kulkarni and N. Anand, "Equal Cost Routes Support for RIP/RIPNG," draft-janardhan-naveen-rtgwg-equalcostroutes-rip-00, June 2007.

[20] R. Chandra, P. Bahl, and P. Bahl, "MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card," *Proc. IEEE INFOCOM,* pp. 882-893. Mar. 2004.

[21] M.K. Marina and S.R. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks," *Proc. Ninth IEEE Int'l Conf. Network Protocols (ICNP),* Nov. 2001.

[22] S.J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC),* pp. 3201-3205. June 2001.

[23] L. Wang, Y. Shu, M. Dong, L. Zhang, and O. Yang, "Adaptive Multipath Source Routing in Ad Hoc Networks," *Proc. IEEE Int'l Conf. Comm. (ICC),* pp. 867-871, June 2001.

[24] Z. Ye, S.V. Krishnamurthy, and S.K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," *Proc. IEEE INFOCOM,* pp. 270-280, Mar. 2003.

[25] D.G. Thaler and C.V. Ravishankar, "Using Name-Based Mappings to Increase Hit Rates," *IEEE/ACM Trans. Networking,* vol. 6, no. 1, pp. 1-14, Feb. 1998.

[26] Y. Lee and Y. Choi, "An Adaptive Flow-Level Load Control Scheme for Multipath Forwarding," *LNCS,* vol. 2093, pp. 771-779, Springer-Verlag, July 2001.

[27] Z. Cao, Z. Wang, and E. Zegura, "Performance of Hashing Based Schemes for Internet Load Balancing," *Proc. IEEE INFOCOM,* pp. 332-341, Mar. 2000.

[28] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP Performance Metrics," RFC 2330, May. 1998.

[29] C. Demichelis and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)," RFC 3393, Nov. 2002.

[30] N.M. Piratla, A.P. Jayasumana, A.A. Bare, and T. Banka, "Reorder Buffer-Occupancy Density and Its Application for Measurement and Evaluation of Packet Reordering," *Computer Comm.,* vol. 30, no. 9, pp. 1980-1993, June 2007.

[31] N.M. Piratla and A.P. Jayasumana, "Reordering of Packets Due to Multipath Forwarding—An Analysis," *Proc. IEEE Int'l Conf. Comm. (ICC),* pp. 829-834, June 2006.

[32] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, and J. Perser, "Packet Reordering Metrics," RFC 4737, Nov. 2006.

[33] A. Jayasumana, N. Piratla, T. Banka, and R. Whitner, "Improved Packet Reordering Metrics," RFC 5236, June 2008.

[34] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP Backbone," *IEEE/ACM Trans. Networking,* vol. 15, no. 1, pp. 54-66, Feb. 2007.

[35] J.C.R. Bennett, C. Partridge, and N. Shectman, "Packet Reordering Is Not Pathological Network Behavior," *IEEE/ACM Trans. Networking,* vol. 7, no. 6, pp. 789-798, Dec. 1999.

[36] G. Iannaccone, S. Jaiswal, and C. Diot, "Packet Reordering inside the Sprint Backbone," Technical Report TR01-ATL-062917, Sprint ATL, June 2001.

[37] D. Loguinov and H. Radha, "Measurement Study of Low-Bitrate Internet Video Streaming," *Proc. First ACM SIGCOMM Workshop Internet Measurements,* pp. 281-293, Nov. 2001.

[38] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "On the Performance Analysis of Traffic Splitting on Load Imbalancing and Packet Reordering of Bursty Traffic," *Proc. IEEE Int'l Conf. Network Infrastructure and Digital Content (IC-NIDC '09),* Nov. 2009.

[39] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-Way Delay Metric for IPPM," RFC 2679, Sept. 1999.

[40] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003.

[41] S. Prabhavat, H. Nishiyama, Y. Nemoto, N. Ansari, and N. Kato, "Load Distribution with Queuing Delay Bound over Multipath Networks: Rate Control Using Stochastic Delay Prediction," *Proc. 26th Int'l Comm. Satellite Systems Conf. (ICSSC),* June 2008.

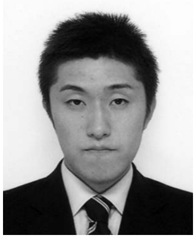
[42] M. Shreedhar and G. Varghese, "Efficient Fair Queueing Using Deficit Round Robin," *IEEE/ACM Trans. Networking,* vol. 4, no. 3, pp. 375-385, June 1996.

[43] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, May. 2000.

[44] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001.

[45] P. Danzig, J. Mogul, V. Paxson, and M. Schwartz, "The Internet Traffic Archive," http://ita.ee.lbl.gov/index.html, Mar. 1995.

Sumet Prabhavat received the BE degree in electrical engineering from Chiangmai University, Thailand, in 1993, and the ME degree in electrical engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2003. Since then, he has been working as a full lecturer in the Faculty of Information Technology at KMITL. Prior to join the Faculty of Information Technology, he was a research assistant in Communication Network Laboratory at the Research Center for Communications and Information Technology (ReCCIT), KMITL. Currently, he is working toward the PhD degree at the Graduate School of Information Sciences (GSIS), Tohoku University, Japan. His main research interests include load distribution, load balancing, performance analysis, application of queuing theory, and congestion control on communication networks. He received the Best Paper Award at the IEEE International Conference on Network Infrastructure and Digital Content (IEEE IC-NIDC) in 2009. He is a student member of the IEEE.



Hiroki Nishiyama received the MS and PhD degrees in information science from Tohoku University, Japan, in 2007 and 2008, respectively. He also worked as a research fellow of the Japan Society for the Promotion of Science (JSPS) for one and a half years since 2007. He has been an assistant professor at the Graduate School of Information Sciences (GSIS), Tohoku University, since Oct. 2008. He received the Best Paper Award at IEEE Global Communications Conference 2010 (GLOBECOM 2010) and at the 2009 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009). He is also a recipient of the 2009 FUNAI Research Incentive Award of FUNAI Foundation for Information Technology (FFIT). He has been engaged in research on traffic engineering, congestion control, satellite communications, ad hoc and sensor networks, and network security. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE), and an IEEE member.

Nirwan Ansari (S'78-M'83-SM'94-F'09) received the BSEE (*summa cum laude* with a perfect gpa) from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the MSEE degree from University of Michigan, Ann Arbor, in 1983, and the PhD degree from Purdue University, West Lafayette, Indiana, in 1988. He joined NJIT's Department of Electrical and Computer Engineering as an assistant professor in 1988, tenured associate professor in 1993, and has been full professor since 1997. He has also assumed various administrative positions at NJIT. He authored *Computational Intelligence for Optimization* (New York: Springer, 1997, translated into Chinese in 2000) with E.S.H. Hou and edited *Neural Networks in Telecommunications* (New York: Springer, 1994) with B. Yuhas. His current research focuses on various aspects of broadband networks and multimedia communications. He has also contributed 350 technical papers, more than one third of which are in widely cited refereed journals/magazines. He was/is serving on the Advisory Board and Editorial Board of eight journals, including as a senior technical editor of *IEEE Communications Magazine* (2006-2009). He had/has been serving the IEEE in various capacities such as the chair of IEEE North Jersey COMSOC Chapter, chair of IEEE North Jersey Section, member of IEEE Region 1 Board of Governors, chair of IEEE COMSOC Networking TC Cluster, chair of IEEE COMSOC Technical Committee on Ad Hoc and Sensor Networks, and chair/TPC Chair of several conferences/symposia. He has been frequently invited to deliver keynote addresses, distinguished lectures, tutorials, and talks. Some of his recent recognitions include an IEEE Fellow (Communications Society, Class of 2009), IEEE Leadership Award (2007, from Central Jersey/Princeton Section), the NJIT Excellence in Teaching in Outstanding Professional Development (2008), IEEE MGA Leadership Award (2008), the NCE Excellence in Teaching Award (2009), a couple of best paper awards, Thomas Alva Edison Patent Award, and designation as an IEEE Communications Society Distinguished Lecturer (2006-2009, two terms). He is a fellow of the IEEE.

Nei Kato received the MS and PhD degrees in information engineering from Tohoku University, Japan, in 1988 and 1991, respectively. He joined Computer Center of Tohoku University in 1991, and has been a full professor at the Graduate School of Information Sciences since 2003. He has been engaged in research on computer networking, wireless mobile communications, image processing, and neural networks. He has published more than 200 papers in journals and peer-reviewed conference proceedings. He currently serves as the chair of IEEE Satellite and Space Communications TC, the secretary of IEEE Ad Hoc and Sensor Networks TC, the vice chair of IEICE Satellite Communications TC, a technical editor of *IEEE Wireless Communications*(2006∼), an editor of *IEEE Transactions on Wireless Communications*(2008∼), an associate editor of IEEE Transactions on Vehicular Technology (2009∼). He served as a coguest editor for *IEEE Wireless Communications Magazine* SI on "Wireless Communications for E-healthcare", a symposium cochair of GLOBECOM '07, ICC '10, ICC '11, ChinaCom '08, ChinaCom '09, and WCNC 2010-2011 TPC vice chair. His awards include Minoru Ishida Foundation Research Encouragement Prize (2003), Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee (2005), the FUNAI information Science Award (2007), the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion (2008), the IEICE Network System Research Award (2009), and best paper awards from many prestigious international conferences such as IEEE GLEBECOM, IWCMC, etc. Besides his academic activities, he also serves as a member on the expert committee of Telecommunications Council, the special commissioner of Telecommunications Business Dispute Settlement Commission, Ministry of Internal Affairs and Communications, Japan, and as the chairperson of ITU-R SG4, Japan. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and a senior member of the IEEE.