東北大学機関リポジトリ
TOUR
Tohoku University Repository

# A Study on Traffic Distribution Models over Multipath Networks



| | |
|---|---|
| 著者 | Prabhavat Sumet |
| 学位授与機関 | Tohoku University |
| URL | http://hdl.handle.net/10097/49877 |

# A Study on Traffic Distribution Models over Multipath Networks

# マルチパスネットワークにおける トラヒック分散モデルに関する研究

A dissertation presented

by

Sumet Prabhavat

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Information Sciences

Graduate School of Information Sciences

TOHOKU UNIVERSITY

January, 2011

# A Study on Traffic Distribution Models over Multipath Networks

# マルチパスネットワークにおける トラヒック分散モデルに関する研究

A dissertation presented

by

Sumet Prabhavat

Approved as to style and content by

Prof. 加藤 寧 (Nei Kato) _____

Prof. 曽根 秀昭 (Hideaki Sone) _____

Prof. 木下 哲男 (Tetsuo Kinoshita) _____

# Abstract

The rapid growth in demand for high-speed and high-quality multimedia and real-time communications has been a major driving force for research and development of a traffic load distribution scheme. An effective model of load distribution becomes essential to efficiently utilize multiple parallel paths for multimedia data transmission and real-time applications. Using multiple paths as a single path with aggregate bandwidth is preferable to provisioning a single large-bandwidth path, since it improves scalability of networks, increases affordability for users, and also provides flexibility in bandwidth management for network operators.

Bandwidth aggregation and network-load balancing are important issues that have attracted tremendous amount of research, and a large number of traffic load distribution approaches have been proposed. At first, we analyze various examples of existing load distribution models, and then compare and identify their exhibited advantages as well as shortcomings, based on a number of significant criteria such as the ability to balance load and to maintain packet ordering, along with several other issues, which affect network performance perceived by users. We present a thorough literature review of various existing load distribution models, and classify them in terms of their key functionalities such as traffic splitting and path selection. The classification and performance analysis of load distribution models conducted in this study provides useful information for further research in this area.

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. In addition, some of them require communication functions leading to network overhead which results in an increase of latency (i.e., packet delay). This dissertation presents a study towards an effective model of load distribution for multimedia data transmission and real-time applications which are commonly known to be sensitive to packet delay, packet delay variation, and packet reordering. To this end, we proposes a new load distribution model, i.e., Effective Delay-Controlled Load Distribution (E-DCLD), 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.

# Acknowledgements

I would like to express my sincere gratitude to my advisor, Prof. Nei Kato, for his helpful guidance and support throughout my PhD research at Tohoku University. Working with him has been my most valuable experience, broadening and enriching my research. In addition, his commitment to excellence and integrity has shaped my perspective.

I would like to express my appreciation to Prof. Nirwan Ansari and Asst. Prof. Hiroki Nishiyama for their kind assistance along the path to graduation. Not only have they taught me a great deal technically but I have found their other lessons equally important.

I would like to acknowledge my thesis committee members, Prof. Hideaki Sone and Prof. Tetsuo Kinoshita, for their interest and for their constructive comments that help to improve this thesis.

Many thanks to the Japanese Ministry of Education for giving me the opportunity to study in Japan by providing me with the scholarship that allowed me to perform the research for this thesis in Tohoku University, one of the best universities in Japan.

Many thanks also to King Mongkut's Institute of Technology Ladkrabang for recommending me to this PhD research program and Assoc. Prof. Ruttikorn Varakulsiripunth for his assistance in the enrollment process.

I would like to thank Motoko Shiraishi and Takako Kase for providing me with all the necessary administrative documents. I would also like to thank for friendly help in both academic and non-academic aspects during my stay in Japan to all friends in Kato laboratory, especially Zubair, Takahashi, George, Furuya, Chaloemchai, and Panu.

Last but not least, I would like to express my appreciation to my parents (Somchai and Malee Prabhavat) as well as my brothers (Pop and Pond) for their continuous love and support; and my girlfriend (Yui) for her encouragement and patience during the years I have been working on this thesis.

# Table of Contents

# List of Tables

Table Page

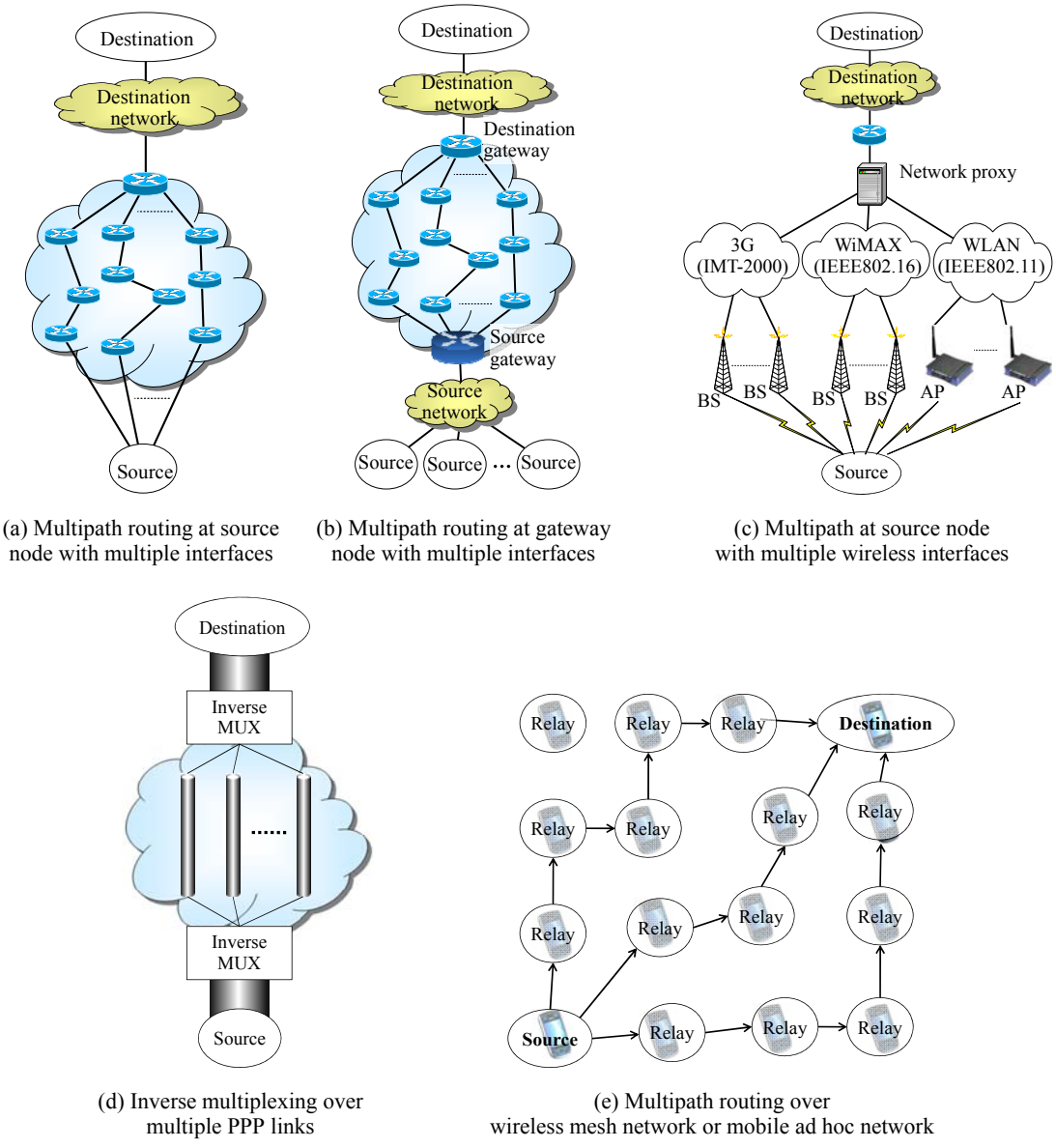
# List of Figures

Figure Page

# Chapter 1

# Introduction

## 1.1. Presences and Benefits of Multipath Environment




(a) Multipath routing at source node with multiple interfaces

(b) Multipath routing at gateway node with multiple interfaces

(c) Multipath at source node with multiple wireless interfaces

(d) Inverse multiplexing over multiple PPP links

(e) Multipath routing over wireless mesh network or mobile ad hoc network

**Figure 1.1.** Examples of various multipath configurations.

The demand for a wide variety of network services has been the major driving force for innovation and development of various networking technologies [1].

### 1.1.1. Presences of Multipath Environment

Multipath configurations can be established in several different ways. Figs. 1.1(a)–(b) present generalized cases where a source or a gateway in the network distributes traffic. While there is just one distribution point for simplicity in Fig. 1.1(b), multiple distribution points can indeed exist between source and destination gateways, and load balancing in such case is referred to as multi-stage load balancing [2]. One of the most well-known routing techniques to establish multiple path routing is Equal-Cost Multi-Path (ECMP) routing [3], [4] which is currently supported by Internet routing protocols such as Open Shortest Path First (OSPF) [5], Routing Information Protocol (RIP) [6], [7], and Enhanced interior gateway routing protocol (EIGRP) [8]. In Multi-Protocol Label Switching (MPLS) networks [9], the source and destination gateways correspond to an ingress and egress router, respectively. The multiple paths between them can be setup by using a signaling protocol, e.g., Constraint-based Routing Label Distribution Protocol (CR-LDP) [10] or Resource Reservation Protocol-Traffic Engineering (RSVP-TE) [11]. In addition, various kinds of dynamic traffic engineering techniques for load balancing over multiple paths, such as [12], [13], and some others overviewed in [14], have been proposed. Fig. 1.1(c) is a special case of Fig. 1.1(a) where the first hop from the source is via a wireless medium. Owing to advances of wireless communications, we can simultaneously use several different types of wireless access networks, e.g., 3G (IMT-2000), WiMAX (IEEE 802.16), and

Wireless Fidelity (IEEE 802.11). On the other hand, inverse multiplexing [15] depicted in Fig. 1.1(d) can be considered as an abstraction of Fig. 1.1(b). It is a popular technique to exploit multiple parallel point-to-point narrowband paths as a single point-to-point broadband path by using the bandwidth aggregation technology [16]. Wide Area Multi-link PPP (WAMP) [17], strIPe [18], and Dynamic Hashing with Flow Volume (DHFV) [19] are implementations of inverse multiplexing. Fig. 1.1(e) presents a generalized model of relay networks such as Mobile Ad hoc Networks (MANETs), wireless mesh networks, and satellite mesh networks. Split Multipath Routing (SMR) [20] and Multi-path Source Routing (MSR) [21] developed based on Dynamic Source Routing (DSR) [22], and Ad hoc On-demand Distance Vector - Multipath (AODVM) [23] and Ad hoc On-demand Multipath Distance Vector (AOMDV) [24] developed from Ad hoc On-demand Distance Vector (AODV) [25] are notable multipath routing protocols for MANETs. For satellite mesh network consisting of non-geostationary satellites, Explicit Load Balancing (ELB) [26] has been developed to distribute traffic among multiple different links in order to avoid traffic convergence.

### 1.1.2. Benefits of Multipath Environment

As mentioned above, the presence of several physical/logical interfaces incorporated with a multipath routing/forwarding protocol allows users to use multiple paths in establishing simultaneous connections. A primary objective of multiple paths is to improve *network reliability* by increasing network availability (i.e., reducing network downtime); a main path was used for data transmission while the other ones were backups which would be activated when the main path became unavailable.

Currently, the exploitation of multiple paths no longer aims only at circumventing single point of failure scenarios but also focuses on facilitating network provision [1], where its effectiveness is indeed essential to maximize high quality network services and guarantee Quality of Service (QoS) at high data rates. Using multiple paths as a single path with aggregate bandwidth is a practical solution which is preferable rather than provisioning a large-bandwidth path because it offers a possibility to establish a very large-bandwidth connection. This improves both *scalability* to support the future growth in bandwidth demand and *affordability* for network users. It also provides *flexibility* in bandwidth management within the communication protocol over the multipath network. Network bandwidth capacity can be controlled by the number of (active) multiple paths combined to a single path: the larger the number of multiple paths, the higher the bandwidth capacity of the network path. The network bandwidth capacity can be adjusted according to the bandwidth demand which can change dynamically over time.

### 1.2. Problems and Motivations

Bandwidth aggregation and network-load balancing are major issues that have attracted a large amount of research, and a number of load distribution approaches have been proposed. Each of the load distribution models exhibits different characteristics, advantages, and drawbacks. A comprehensive review of the existing schemes is necessary for research in this area.

Demands for network infrastructure in providing high-speed high-quality network services that can support them have been continuously growing, since

multimedia and real-time applications which are commonly known to be sensitive to delay have been very popular network applications. Network capacity provisioning and QoS guarantees become major issues in meeting this demand. Some load distribution approaches working well for traditional best-effort applications are no longer suitable. Therefore, an effective delay-controlled load distribution is critical to efficiently utilize multiple available paths for multimedia data transmission and real-time applications.

### 1.3. Objectives

First, we survey existing load distribution models in previous works, identify their advantages as well as shortcomings, and analyze the exhibited characteristics. Based on the analysis of existing models obtained from the survey, we propose an effective model of load distribution that is essential to efficiently utilize multiple parallel paths for multimedia data transmission and real-time applications.

### 1.4. Contributions

The work presented in this dissertation provides two main contributions. The first contribution is a comprehensive review of existing load distribution models, which presents useful information for research in this area, e.g., collection, classification, performance issues, analysis, and comparison of existing load distribution models in previous works. The other contribution is an effective model of load distribution, i.e., Effective Delay-Controlled Load Distribution Model (E-DCLD), which can effectively reduce latency (and variation of latency) to successfully transmitting a packet, without

incurring network overhead. The latency in the focus of this work is the end-to-end delay in transmitting a packet and the additional time required in reordering the packet.

### 1.5. Organization of the Dissertation

The rest of this dissertation is organized as follows. Chapter 2 describes and classifies existing load distribution models in terms of internal functions, i.e., traffic splitting and path selection components. Chapter 3 describes important performance issues in load distribution and conducts performance comparisons in various criteria of existing load distribution models. Chapter 4 identifies delay-related problems caused by existing load distribution models; then proposes Effective Delay-Controlled Load Distribution (E-DCLD) model which can effectively mitigate packet delay, packet delay variation, and packet reordering. E-DCLD is evaluated and compared with the current existing load distribution models under various traffic conditions. Chapter 5 concludes the main advantages of the work presented in this dissertation.

# Chapter 2

# Survey on Load Distribution Models

## 2.1. Generalized Multipath Forwarding Mechanism

The important role of load distribution is engineered by the traffic splitting and path selection, which are the key components of multipath forwarding. After having described the general multipath forwarding mechanism, different types of traffic units and different path selection schemes will be discussed.

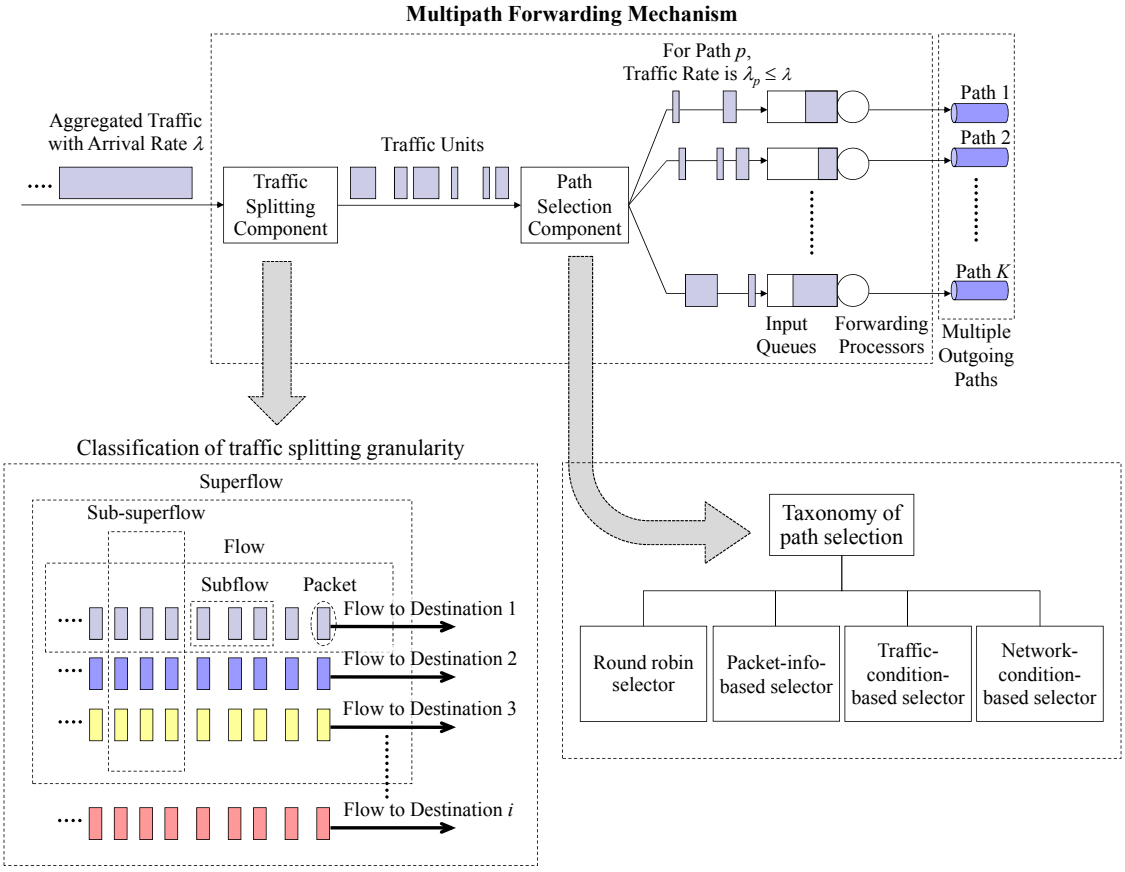



**Figure 2.1.** Functional components of the multipath forwarding mechanism and classifications of internal functional components.

Fig. 2.1 illustrates the functional components of multipath forwarding: traffic splitting and path selection. The traffic splitting component splits the traffic into traffic units, each of which independently takes a path which is determined by the path selection component. If the forwarding processor is busy, each traffic unit is queued in the input queue attached at the output link as determined by the path selection. Various multipath forwarding models perform load distribution in different manners. Each model exhibits different advantages and shortcomings because of the difference in their internal functional components, i.e., traffic splitting and path selection. Note that the input queue and forwarding processor components do not assume further roles in load distribution.

### 2.1.1. Traffic Splitting Component

By the traffic splitting component, aggregated traffic from traffic sources is split into several traffic units, where the constitution of a traffic unit depends on the level of splitting granularity. The traffic splitting classification is illustrated in Fig. 2.1.

In Packet-level traffic splitting, traffic is split into the smallest possible scale, i.e., a single packet. Path selection is individually decided for each packet. A load distribution model with this kind of traffic splitting is referred to as a packet-based load distribution model.

In Flow-level traffic splitting, packet-identifiers, which are determined from destination addresses stored in packet headers, are taken into consideration in splitting. All packets heading for the same destinations are grouped together because of their similar packet-identifiers; the group is defined as a unit of flow, where the flow

identifier is a unique identifier of each flow. Splitting traffic at this level can maintain packet ordering since path selection for all packets in the same flow is identical. The path selection is independent from flow to flow. A load distribution model with this kind of traffic splitting is referred to as a flow-based load distribution model. To further specify a particular flow, for example, the following packet header information can be used [27]: source address, type of service, protocol number, and so on. Taking a source address, type of service, and protocol number into account in a splitting condition allows each flow to be differentiated by its source, class of service, and type of network application, respectively.

In Subflow-level traffic splitting, a flow of packets heading for the same destination is allowed to be split into a traffic unit of subflow (i.e., a subset of packets in an original flow), sometimes referred to as a flowlet. All packets in a subflow are destined for the same destination, but all packets heading for the same destination may be carried in different subflows. Various flow-characteristics can be taken into account in a splitting condition, e.g., packet inter-arrival time and packet arrival rate, depending on the load balancing objective. Reference [28] shows an example of the splitting condition to achieve a specific load balancing objective, which will be described in the next section.

In Superflow-level traffic splitting, traffic is split into superflows, each of which is a group of flows having the same result calculated from their flow identifiers by some specific function. As compared to a flow-level traffic splitting, packets heading for different destinations can be grouped into the same superflow. A hash function is a

well-known example used in the Internet load balancing. A traffic splitting scheme that uses a hash algorithm to generate hash values of packet identifiers is typically known as a hash-based traffic splitting scheme [29].

In Sub-superflow-level traffic splitting, a sub-superflow is a group of packets (which is a subset of a superflow) which satisfy a certain splitting condition, similar to the relation between subflow and flow. As compared to a subflow, some packets in a sub-superflow head for different destinations, but have the same hashing result of their packet identifiers. In addition to characteristics of each flow, those of aggregated flows (e.g., flow inter-arrival time and the number of flows in a sub-superflow) can be taken into account in the traffic splitting.

### 2.1.2. Path Selection Component

The path selection component is responsible for choosing a path for an arrived packet. Path selection for each of the traffic units is independently decided. If the scale of traffic unit is a single packet, each of the arrived packets is treated independently while, if a traffic unit has a larger scale than a single packet such as flow, subflow, superflow, and sub-superflow, all packets of the same traffic unit will be treated in the same manner. Most path selection schemes can be categorized into four types as shown in Fig. 2.1 and described as follows.

Round robin selector (RR) is a path selection scheme in which successive traffic units are sent across all parallel paths in a round robin manner. RR selector [30], [31] is rather simple, has the computational complexity of $O(1)$, and requires no additional network information for path selection.

In Packet-info-based selector (Packet-info), a packet identifier obtained from packet header information of an arrived packet plays an important role in the path selection. Typically, an outgoing path is determined based on the output of a function of the packet identifier (e.g., a mapping function and a modulo-N hashing function). If a hash function is used, it is known as the hash-based path selection mechanism.

In Traffic-condition-based selector (TrafficCon), traffic conditions are taken into account in path selection. They include traffic load, traffic rate, traffic volume, and the number of active flows [32], and are selected depending upon control objectives.

In Network-condition-based selector (NetCon), network condition is used to determine the outgoing path, such as path delay, path loss, and backlogged queue length of the path, or outgoing link are used to determine the output path, according to the goal of load balancing. Shortest-Path-First (SPF) and Least-Loaded-First (LLF) [64], [65], [66] are some of the most well known path selection schemes. In SPF, a path with the lowest cost will be selected for an arrived packet. In LLF, a path having the smallest load or the shortest queue will be selected instead.

## 2.2. Classifications and Descriptions of Existing Models

Existing load distribution models can be classified into two categories, namely, non-adaptive and adaptive models. In addition, they may be further classified based on their required additional information for distributing load such as info-unaware, packet-info-based, traffic-condition-based, and network-condition-based information, as illustrated in Fig. 2.2. Various examples of load distribution models are investigated in terms of their functionalities, characteristics as well as internal functional components.

The first subsection presents info-unaware models that make a raw decision on distributing traffic without taking external information into account and (non-adaptive) packet-info-based models that require packet information obtained from the packet header. Adaptive models requiring traffic condition estimated from the incoming traffic and network condition measured by network measurements will then be presented in the subsequent section.

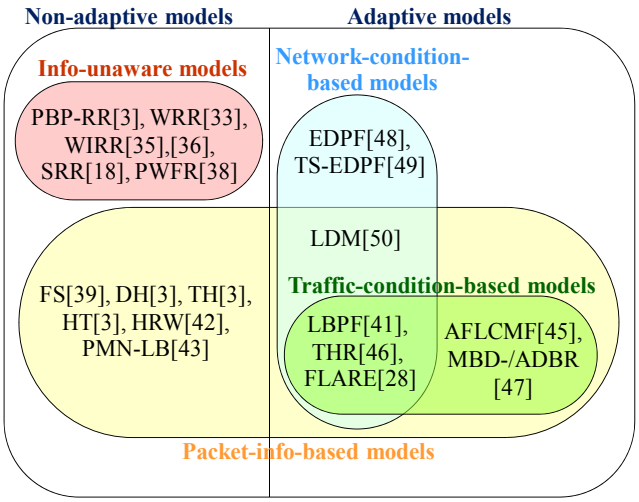



**Figure 2.2.** Load distribution model classification.

### 2.2.1. Non-adaptive Models

"Info-unaware" refers to the class of models which make a raw decision on distributing traffic without taking external information into account, and "packet-info-based" refers to the class of models that require packet information obtained from the packet header. Their advantages/limitations are summarized in Table 2.1 and described as follows.

**Table 2.1.** Summary of non-adaptive load distribution models.

| Model | Advantages and enhancement | Remaining problems and limitations |
|---|---|---|
| **Info-unaware Models** | | |
| PBP-RR [3] | Simple. No communication overhead. | Not applicable for multiple paths with different characteristics. No mechanism to prevent packet reordering. |
| WRR [33] | Ability to control the amount of load among outgoing paths. | Variation in packet size distribution may affect load balancing efficiency. No mechanism to prevent packet reordering. |
| WIRR [35], [36] | Prevent the continuous use of a particular path. | Similar to WRR. |
| SRR [18] | Similar to WRR, but byte-based deficit counter allows to cope with variation in packet size distribution. | No mechanism to prevent packet reordering. |
| PWFR [38] | Only the path with the largest deficit load is chosen; this helps decrease load balancing deviation. | Similar to SRR. |
| **Packet-info-based (non-adaptive) Models** | | |
| FS [39] | The number of flows can be uniformly distributed among paths. | Cache memory is required to store flow-path mapping entry. Load imbalance caused by variation in flow size distribution. |
| DH [3] | Simple. No communication overhead. | Load imbalance caused by variation in flow size distribution and non-uniformity of hash distribution. High disruption. |
| TH [3] | Load sharing ratio can be controlled by customizing a mapping table between a path and a group of flows, i.e., superflow. | A superflow tends to have a large variation in traffic-unit size distribution, leading to load imbalance. |
| HT [3] | Load sharing ratio can be controlled, similar to TH. Degree of disruption can be reduced up to 75%, as compared to TH. | Similar to TH. |
| HRW [42] | Degree of disruption is minimized, i.e., only one path is affected by a change of path state. | As compared to DH, TH, and HT, higher complexity; and poorer lookup performance. |
| PMN-LB [43] | Low disruption and low complexity. | Load imbalance caused by variation in flow size distribution and non-uniformity of hash distribution. |

#### 2.2.1.1. Info-unaware Models

Load distribution models requiring no information regarding traffic and network condition are classified into the info-unaware class; they do not require collecting any information on traffic load or from the network. A common major drawback of models in this class is inability to maintain packet ordering. Some additional mechanism is

required to preserve packet ordering, e.g., synchronization recovery [18] and packet reordering recovery.

Packet-By-Packet Round-Robin (PBP-RR)

PBP-RR has been implemented in several applications, e.g., ECMP routing and inverse multiplexing. The first example is incorporated in packet-switched networks while the latter is in multiple point-to-point networks. Since PBP-RR implements the packet-based round-robin path scheduling [3], it achieves simplicity and starvation-free (i.e., no idle path exists while a packet is waiting to be sent) and requires no communication overhead; however, inability to maintain per-flow packet ordering and to control the amount of load shared (by the multiple paths) are its drawbacks. Owing to its inability to control the amount of shared load, PBP-RR is not able to balance load among heterogeneous multiple paths. If the parameter of each path is different (their bandwidths are unequal), PBP-RR can cause problems such as over-utilization of a path with low capacity and under-utilization of a path with high capacity.

Weighted Round Robin (WRR)

The idea of weighted sharing by using WRR path scheduling [33] is implemented to support heterogeneous multiple paths [8], [34]. Each path is assigned a value that signifies, relative to the other paths in the set of multiple paths, how much traffic load should be assigned on that connection path. This "weight" determines how many more (or fewer) packets are sent via that path as compared to other paths. In other words, the numbers of packets assigned to paths are limited by weights of the paths.

WRR has been incorporated in several routing protocols such as EIGRP [8] and MSR [21]. In WRR, load imbalance can occur due to variation in the size of packets. Also, it can occur due to improper weight assignment (i.e., a path with low bandwidth is assigned a large weight while a path with large bandwidth assigned a low weight).

Weighted Interleaved Round Robin (WIRR)

WIRR [35], [36] possesses characteristics almost similar to WRR except that a successive packet will be sent to the next parallel path in a round robin manner. Only the paths having a smaller number of sent packets than the desired number will remain in a pool (of paths which can be selected) for the next round. Unlike WRR, WIRR prevents continuous use of a particular path; it can thus reduce non-work-conserving idle time (i.e., duration time when a particular path is idle while a packet is waiting to be sent). Similar to the problem stated in the case of RR, both WRR and WIRR schemes are still unable to maintain per-flow packet ordering.

Surplus Round Robin (SRR)

SRR has been implemented for load balancing in packet-switched networks, as a part of strIPe protocol [18]. SRR is based on a modified version of Deficit Round Robin (DRR) [37], which is a modified WRR. 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. SRR uses a byte-based deficit counter. At the beginning of each round, the deficit counter is increased by the given (positive) quantum 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 positive deficit counter will be selected in a round robin manner. If the deficit counters of all paths are non-positive, the round is over and a new round is begun. With varying packet sizes, PBP-RR, WRR, and WIRR result in unfair sharing in favor of longer packets; SRR has a better performance in load balancing because it uses a byte-based counter, and it is thus not affected by packet-size variation.

Packet-by-packet Weighted Fair Routing (PWFR)

PWFR [38] is designed aiming to effectively perform load sharing and outperform a widely used scheme such as RR in multipath packet-switched networks. In PWFR, each path has a given routing weight indicating the amount of desired load, where the term “load” is the number of bytes of a packet. For each packet arrival, the deficit counter of each path is increased by a fraction of the packet size for that path. A path with the maximum value of the deficit counter is selected for forwarding the packet; then, its deficit counter is decreased by the packet’s size. As compared to round robin based models, it can minimize load balancing deviation (i.e., the difference between the desired and actual loads); it is a deterministically fair traffic splitting algorithm which is useful in the provision of service with guaranteed performance in a network with multiple paths. However, it has computational complexity of $O(n)$; processing time of the path selection for each packet increases when the number of paths increases. In a large and high speed network, a high performance processor is necessary.

#### 2.2.1.2. Packet-info-based (Non-adaptive) Models

The inability to prevent packet reordering is the major problem of the info-unaware models. Since the packet reordering problem can be completely mitigated by selecting the same path for packets heading for the same destination, network-related packet information (e.g., destination address, source address, and so on) is required for path selection. This idea has been incorporated in [39] and has also been studied in hash-based schemes [4], [29], [40].

Fast Switching (FS)

FS [39] is a flow-based model with Packet-info-based and RR path selection schemes, implemented in fast-switching which is a Cisco-proprietary technology. In the same flow, packets are sent via the same path as the preceding ones unless the buffer runs out of space. When a new flow emerges, packets belonging to the new flow will be sent via the next parallel path in a round robin manner and a new flow-path mapping entry is stored in a cache memory. Different from hash-based schemes, the flow is not permanently pinned to a particular path by hashing the flow identifier; the number of flows can thus be uniformly distributed among the paths. However, FS cannot deal with skewness of flow size distribution. Moreover, FS requires memory to store the flow state, where the number of active flows can grow infinitely. When a new flow emerges while there is no available memory space, the oldest flow-path mapping is replaced by the new mapping record entry. As a consequence, the path for the oldest flow may change. Insufficient memory space to store the state information can thus result in packet reordering problems. It is essential for the memory space to be large

enough to hold the flow-path mapping record, and to ensure that the record will not be replaced before the preceding packet arrives at its destination. This allows the current packet to be sent via a different path without the risk of reordering. In this sense, a path for forwarding the packet is determined by looking up in a flow-path mapping table, resulting in the computational complexity of $O(n)$, where $n$ is the number of entries in the flow-path mapping table. This can create scalability issues when the number of flows or paths increases. In FS, when a path is removed, all flows mapped to the path become free; they are then treated as new flows. Since only flows mapped to the deleted path are remapped to new paths, the ratio between the number of re-routed flows and the total number of flows in all paths, referred to as the degree of disruption [4], [40], is at the minimum level, $1/K$.

Direct Hashing (DH)

DH is a conventional flow-based model which is widely deployed in multipath routing protocols [3], [4], [40]. It performs hash-based load balancing for ECMP routes. Its functional components are illustrated in Fig. 2.3(a). To obtain the outgoing path, it executes modulo-$K$ hash algorithm: taking the packet identifier, $X$, (obtained from packet information such as destination address), applying a hash function, $h(X)$, and taking modulo of the number of multiple paths, $\mathrm{mod}(h(X), K)$. Having a simple algorithm with the computational complexity of $O(1)$ and having no communication overhead are its advantages. However, performance in load balancing of DH depends on the distribution of hash values. When all flows have the same value of the hashed flow ID and so all packets are forwarded via a single path, this will result in the worst

load imbalance. Moreover, DH cannot deal with the variation of the flow size distribution; skewness of the flow size distribution inherent in the network environment has a significant impact on its performance in load balancing. DH can achieve the best balancing performance when hashing results and flow sizes are uniformly distributed [29], [41]. The other drawback of DH is that a number of flows are redistributed when a path is added or removed since a change in the value of $K$ is likely to cause a different result of mod($h(X)$, $K$); the degree of disruption is large, 1-1/$K$.

Table-Based Hashing (TH)

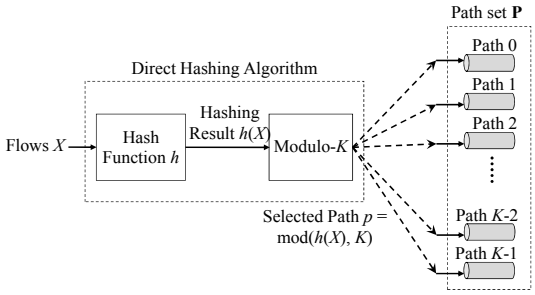
TH [3] is a hash-based load balancing scheme in ECMP routing. Its functional components can be illustrated in Fig. 2.3(b). Each superflow associated with a corresponding bin is assigned to a particular path, according to the bin-to-path mapping table, $f$. The bin involves flows having the same value of the hashed flow ID. TH allows us to distribute traffic in a pre-defined ratio by modifying the allocation of the bins to paths, $f$ [29]; when the mapping is one-to-one, TH corresponds to DH. That is, the load sharing ratio can be controlled by customizing the mapping table. Load imbalance can occur because a superflow has a large variation in superflow size distribution. TH has the computational complexity of $O(1)$, has no communication overhead, and cannot deal with variation of flow size distribution. TH has also poor disruption behavior, 1-1/$K$.

Hash Threshold (HT)

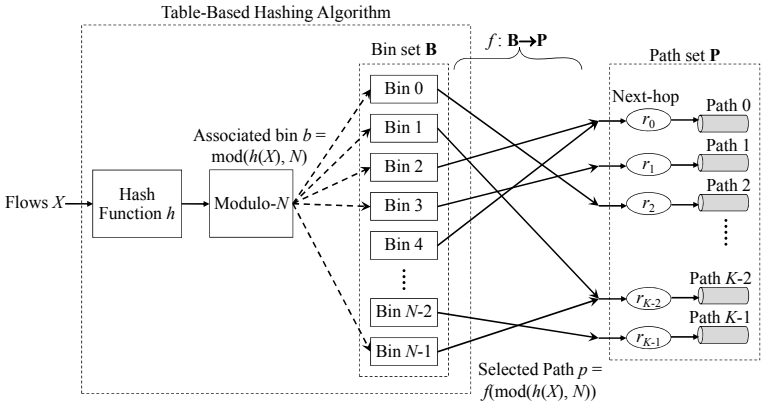
HT [3], which is a load balancing scheme incorporated in ECMP routing, possesses characteristics almost similar to those of TH in Fig. 2.3(b) except the mapping table (*f*). It partitions the hash result space into several regions. Load ratios among multiple paths are controlled by allocating the corresponding region according to the desired ratio; probability of each path selected is determined by the region size [4], [40]. For example, in order to achieve equal load sharing, each region is equally partitioned. A path supposed to be selected for an arrived packet can be determined by finding out which region contains the hashing result of the arrived packet. This can be obtained by rounding up the division of the hashed result by the region size, where the region size can be calculated from the division of the key-space size by the number of multiple paths. HT has the degree of disruption between $0.25+0.25/K$ to 0.5. As compared to TH, HT can improve disruption.

Highest Random Weight (HRW)

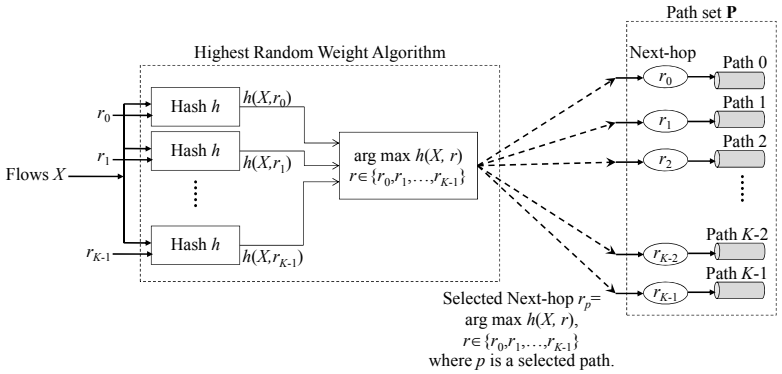
HRW [42] is a load balancing scheme used in WWW caches and in ECMP routing. In HRW, a path is selected based on its random weight computed based on the packet identifier ($X$) and the next hop address ($r_p$) of path $p$; only a path with the highest random weight is selected, as illustrated in Fig. 2.3(c). When an existing path becomes unavailable, only flows mapped to the path are re-routed to the other path with the highest (re-computed) random weight. As compared to DH and TH, HRW can reduce the degree of disruption to the minimal value of $1/K$ [4], [40], but it has a higher computational complexity, $O(n)$. Lookup performance will degrade when the number of flows grows large.




(a) Direct hashing (DH) scheme




(b) Table-Based hashing (TH) scheme




(c) Highest Random Weight (HRW) scheme

**Figure 2.3.** Functional components of the well-known hash-based algorithms for Internet load balancing.

Primary Number Modulo-N Load Balance (PMN-LB)

PMN-LB [43], [44] uses two path selection algorithms: primary and secondary algorithms. The primary algorithm is ordinary modulo-*N* hash algorithm (similar to that of DH). For all flows, the primary algorithm is executed in path selection. However, when the number of available paths changes, it is possible that, without updating the divisor *N*, the ordinary modulo-*N* hash algorithm cannot select available paths for some flows (because the paths selected for them are not available). If this happens, the secondary algorithm will be executed to ensure selection of an available path for the flows. Among available paths, the path indexed by the remainder of flow ID divided by a maximum prime number (not exceeding the number of available paths) is selected. Therefore, only some (not all) flows are affected by an increase or decrease of available paths. Degree of disruption, which depends on the number of paths, is between 0.14 and 0.54 for 8 multiple paths, and between 0.07 and 0.61 for 16 multiple paths. As compared to HRW, PMN-LB provides better lookup performance, $O(1)$, but has a higher degree of disruption. However, the disruption caused by PMN-LB is considered insignificant as compared to the conventional models such as DH, TH, and HT.

### 2.2.2. Adaptive Models

Distributing load in info-unaware models and in packet-info-based (non-adaptive) models cannot efficiently balance load under dynamic conditions of traffic and network which cannot be estimated in advance, e.g., variation of traffic flow, emergence of highly skewed flow-size distribution, and network congestion. Adaptive load distribution can be used to tackle the problems. We further classify adaptive load

distribution models into two classes according to the respective type of conditions. Their advantages and limitations are summarized in Table 2.2 and described as follows.

**Table 2.2.** Summary of adaptive load distribution models.

| Model | Advantages and enhancement | Remaining problems and limitations |
|---|---|---|
| **Traffic-condition-based Adaptive Models** | | |
| AFLCMF [45] | Load sharing ratio can be controlled by a predetermined parameter. | Since adaptation is invoked for all packet arrivals, it can cause flow redistribution and packet reordering. |
| MBD-/ADBR [47] | Redistributing each of excessive loads of over-utilized paths gradually but frequently can decrease load balancing deviation. | Repeating the reassignment processes several times (in each control phase) causes high complexity and increases flow redistribution and packet reordering. |
| **Network-condition-based Adaptive Models** | | |
| EDPF [48] | Selecting a path having the smallest delay can reduce end-to-end delay. | Selecting a path having the smallest delay can cause a risk of packet reordering. |
| TS-EDPF [49] | Scheduling packets on each path based on time slot related to bandwidth negotiated from a QoS server can reduce packet delay and guarantee QoS. | Similar to EDPF. |
| LDM [50] | The shortest path with low utilization has a high precedence to be selected for a new flow. | Load imbalance caused by variation in flow size distribution. |
| **Traffic and Network-conditions-based Adaptive Models** | | |
| LBPF [41] | Splitting only aggressive flows can balance load while causing less flow disruption and packet reordering. | Cannot mitigate load imbalance caused by several non-aggressive flows. |
| THR [46] | By conditional splitting based on flow size and packet inter-arrival time, load balancing can be achieved at the expense of packet reordering (or vice versa). | The optimal point of trade-off between balancing load and preserving packet order is difficult to be determined for a given network condition. |
| FLARE [28] | Considering packet inter-arrival time and path delay in conditional splitting allows balancing load while preventing packet reordering. | Active estimation technique to measure the delay difference causes network overhead and reduction of available bandwidth for users. |

#### 2.2.2.1. Traffic-condition-based Adaptive Models

Load distribution models in this class can adapt to traffic condition including the amount of traffic load (in packets or bytes) as well as traffic characteristics. Information of traffic condition can be collected from input traffic; it does not incur additional network overhead. For highly skewed flow size distribution, traffic load cannot be

balanced by info-unaware models or packet-info-based (non-adaptive) models. Adaptive path selection based on traffic condition can mitigate this problem by selecting the path with high bandwidth to carry a large flow [45]. Splitting traffic flows is another solution. However, splitting all traffic flows can cause a number of re-routed flows. Adaptive traffic splitting which splits only some flows can reduce the number of re-routed flows dramatically [41]. Moreover, conditional splitting only a traffic flow having its packet inter-arrival time larger than some threshold can mitigate the packet reordering problem [46].

Adaptive Flow-level Load Control scheme for Multipath Forwarding (AFLCMF)

Lee and Choi [45] proposed AFLCMF for load balancing in packet-switched networks. When the load ratio for each path (i.e., the load of this particular path over the total load of all paths) is given, the aggregated traffic is split to satisfy the pre-defined load ratio of each path. Each flow, which is classified based on its packet arrival rate, is sent via a path selected corresponding to its class. For example, a flow with rate higher than certain threshold will be sent via path 1; otherwise, it will be sent via path 2. Varying the rate threshold in the flow classification affects the number of flows sent via each path, and thus controls the ratio of load among the multiple paths. To maintain the load ratio, AFLCMF attempts to adjust the rate threshold according to the measured load. Since load assigned on each path is adapted to dynamic changes of the traffic condition, load imbalance caused by the variation of flow size distribution can be mitigated. However, by adjusting to the traffic condition, several flows can experience changes of class, thus resulting in path switching. The re-routed flows are

considered to be disrupted by the adaptation and likely to experience packet reordering. Processing times of flow classification and path selection, with computational complexity of $O(n)$, increase when the numbers of active flows and parallel paths increase, respectively.

Progressive Multiple Bin Disconnection with Absolute Difference Bin Reconnection (MBD-/ADBR)

MBD-/ADBR [47] is a variant version of TH. In contrast, the flow-to-path mapping table *f* illustrated in Fig. 2.3(b) can be dynamically changed. The number of packets in each superflow is taken into account in determining the size of the superflow and the status of the path. The actual load which is the total number of packets forwarded via each path is used to determine whether the path is over-utilized or under-utilized. Each control phase consists of two steps. In the first step, one of the smallest superflows assigned to the most over-utilized path is removed, and thus becomes a free superflow. This step is repeated until all over-utilized paths are under-utilized. The second step is to assign the largest (free) superflow to the most under-utilized path, repeatedly until no free superflow remains. Redistributing excessive load of over-utilized paths, gradually but frequently, can improve load balancing efficiency but cause a number of re-routed flows as well as the risk of packet reordering. MBD-/ADBR has computational complexity of $O(n)$. In each control phase, processing time increases as the numbers of superflows and paths increase.

#### 2.2.2.2. Network-condition-based Adaptive Models

For the models in this class, network conditions such as utilization and delivery time are taken into consideration in path selection.

Earliest Delivery Path First (EDPF)

EDPF [48] was proposed for load balancing in wireless packet-switched networks, and to be implemented in devices (i.e., a mobile host or a network proxy) equipped with multiple interfaces. The corresponding interface will be activated when a path is selected. The goal of EDPF is to ensure that packets reach their destination within certain duration by scheduling packets based on the estimated delivery time. EDPF considers the path characteristics such as delay and bandwidth between the source and destination, and schedules packets on the path which will deliver the packet at the earliest to the destination. Time to finish the transmission is calculated from path delay, time to wait until a path is available, and packet transmission time. The waiting time in the second term can be estimated by tracking the corresponding input queue. The packet transmission time is calculated from the link speed. As compared to other round robin approaches, EDPF achieves better load balancing performance and can reduce packet delay. Load balancing deviation of EDPF is bounded by the maximum packet size, that of SRR is bounded by twice of the maximum packet size, and that of WRR can grow without bound. However, for a packet, selecting a path having the smallest delay poses the risk of packet reordering. In EDPF, the path selection algorithm has computational complexity of $O(n)$; processing time of the path selection increases when the number of paths increases.

Time-Slotted Earliest Delivery Path First (TS-EDPF)

TS-EDPF [49], which is an enhanced version of EDPF, aims to provide manageability for a QoS server in bandwidth allocation for each Mobile Station (MS) in order to reduce the waiting time of packets queued at the Base Station (BS). TS-EDPF modifies the scheduling algorithm in deciding the path selection. Since the available time of each path (i.e., the available time of BS) is divided into time-slots, each of which has a smaller length, the waiting time for the next available time can be reduced. Moreover, TS-EDPF includes the time-slot assigned to an MS on each interface in the estimation of the delivery time of each packet. Before the MS associates with a BS, it negotiates the service level with BSs. Based on the decision from the QoS server, each BS allocates a suitable time-slot to the MS; the waiting time (in a BS queue) of the scheduled packets for their turns to be transmitted can thus be significantly reduced. Therefore, TS-EDPF can reduce packet delay and guarantee quality of service. The scalability of TS-EDPF is similar to that in EDPF.

Load Distribution over Multipath (LDM)

LDM [50] is a load distribution model relying on the traffic engineering concept [51], designed for MPLS networks [9]. LDM is a flow-based model with LLF and SPF path selection schemes. For each arrived flow, path utilization at the moment, in addition to the hop-count of the path, is used to determine the probability of selection of each path; LDM randomly selects a path from several candidates accordingly. In this sense, path utilization and hop count are used as parameters to compute the probability of the particular path to be selected such that a lower utilized and smaller hop-count

path has a higher probability to be selected. However, since LDM does not split a flow, load balancing performance can be degraded by variation in flow size distribution. LDM has computational complexity of $O(n)$; processing time of the path selection for a new flow increases when the number of paths increases.

#### 2.2.2.3. Traffic-condition and Network-condition-based Adaptive Models

For the models in this class, both traffic conditions (e.g., packet inter-arrival time) and network conditions such as utilization and delay are taken into account in traffic splitting and path selection in order to improve the load distribution performance such as load balancing [41], [46], and packet order preservation [28].

Load Balancing for Parallel Forwarding (LBPF)

W. Shi, *et al.* [41] investigated the load imbalance problem caused by the inability of hash-based load balancing schemes in dealing with skewness of flow size distribution of Internet traffic. LBPF [41], a proposed solution for the problem, is an adaptive load balancing scheme that aims to cope with load imbalance due to highly skewed flow size distributions. In the ordinary mode, LBPF selects the path for a flow according to a hashed result of the flow's ID, similar to the conventional hash-based models. In addition, LBPF takes into account the traffic rate of each flow. Relatively high-rate flows can be detected by measuring the number of packets of each flow and comparing to that of the other flows in an observation window (which is the time duration until the total number of counted packets reaches a predefined number). The high-rate flows are classified into a group of aggressive flows. When the system is

under some specific condition (e.g., the system is unbalanced), the adaptation algorithm will be activated. In such condition, each passing packet is checked; if it belongs to one of the aggressive flows, the packet is set to be forwarded via the path with the shortest queue at the moment. In this sense, the aggressive flows which can cause load imbalance are split into several subflows, thus resulting in smaller variation of flow size distribution. That is why LBPF can deal with the skewness of flow size distribution and improve load balancing performance; however, it cannot cope with load imbalance resulting from non-aggressive flows. Moreover, since only the aggressive flows are re-routed, LBPF produces only a small disruption and causes less packet reordering. Note that LBPF does not have an extra preventive mechanism to mitigate packet reordering; packet reordering still occurs. For each packet, processing times of flow classification and path selection algorithms, with computational complexity of $O(n)$, increase as the numbers of active flows and parallel paths increase, respectively.

Table-Based Hashing with Reassignment (THR)

THR [46] is similar to TH but the flow-to-path mapping table *f* illustrated in Fig. 2.3(b) can vary dynamically. In each superflow, a counter and a timer are used to record the number of packets and the packet inter-arrival time, respectively. The actual load, which is the total number of packets forwarded via each path, is used to determine whether the path is over-utilized or under-utilized. In each control phase, one of the superflows assigned to the most over-utilized path is moved to the most under-utilized path (having a small queue-length) by updating the flow-to-path mapping table, accordingly. THR has a pre-determined key parameter, $\beta$, which determines the priority

between improving load imbalance and preventing packet reordering. With $\beta \to 0$, THR aims to reduce the load imbalance by moving the largest superflow. On the other hand, with $\beta \to \infty$, THR focuses more on the packet inter-arrival time to mitigate the packet-reordering problem by moving the superflow with the longest (packet) inter-arrival time. Based on the value of $\beta$, THR can switch its functionality. However, it is difficult to determine the optimal point of trade-off between balancing load and preserving packet order for a given network condition. THR has computational complexity of $O(n)$; in each control phase, processing times of bin and path selection algorithms increase when the numbers of bins and paths increase, respectively.

Flowlet Aware Routing Engine (FLARE)

FLARE [28] was proposed to achieve load balancing while preventing packet reordering, for load distribution among multiple paths in packet-switched networks. In FLARE, a flow is split into several subflows, each of which is referred to as a flowlet. The pre-determined key parameter of FLARE is an inter-arrival time threshold. In this sense, the flowlet can be considered as a group of packets having their inter-arrival time smaller than the threshold. A packet arrived within duration less than the threshold is part of an existing flowlet and will be sent via the same path as the previous one. Otherwise, the packet arrived beyond the threshold corresponds to the head of a new flowlet, and is assigned to a path with the largest amount of deficit load. Path selection of FLARE is approximately similar to that of PWFR; it has computational complexity of $O(n)$; processing time of the path selection for a new flowlet increases when the number of paths increases.

## 2.3. Summary

A number of load distribution models have been proposed in literature. Each model has significant impacts on network performance; it may facilitate high bandwidth connectivity and efficiently utilize multiple network interfaces while its limitations may degrade network performance. Differences in characteristics of each model lead to different advantages and shortcomings.

This section provides overviews of various existing load distribution models. Each model is described in terms of its internal functions in multipath forwarding mechanism, i.e., the traffic splitting and the path selection, which plays an important role in the load distribution. Aggregated traffic can be split into several levels. With a different traffic splitter, a load distribution model exhibits a different characteristic; splitting traffic into single packets allows the load distribution to achieve load balancing, while splitting traffic into flows allows the load distribution to maintain packet ordering. We expect that the classifications and analysis of the internal components will provide a comprehensive understanding of various load distribution models over multiple paths.

Corresponding to the particular internal components, various examples of load distribution models are classified into four different classes, namely, info-unaware, packet-info based, traffic-condition based, and network-condition based models. The *info-unaware models* are load distribution models, which have low complexity, and they do not incur operational cost to the considered network. The *packet-info-based models* can maintain the order of arrived packets. The *traffic-condition-based adaptive*

*models* require information regarding traffic load in making a decision on traffic splitting and path selection. Load sharing can be more precisely controlled in such models. In addition, some models exploit the knowledge of traffic conditions so that only the large flows are split. These particular models can mitigate packet reordering and flow disruption. On the other hand, the *network-condition-based adaptive models* allow load distribution to adapt to network conditions. Based on knowledge of network conditions, with some specific objective, a traffic splitter can split a flow and a path selector can select a path conditionally. With the knowledge of path utilization, a path can be selected appropriately. With the knowledge of path delay, traffic splitter can decide to split only a flow under proper conditions. The performance issues which are mentioned above will be further described in the next section.

# Chapter 3

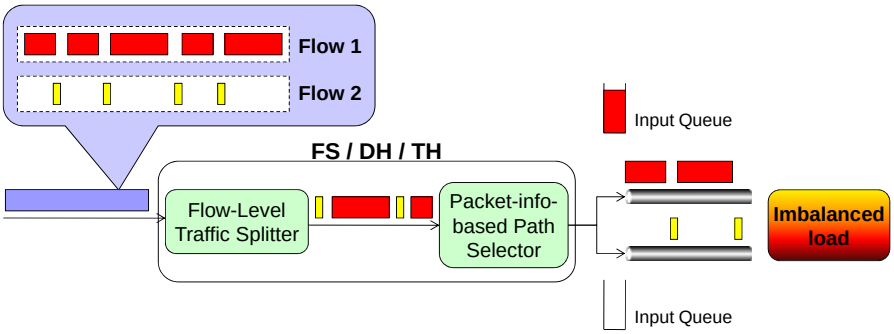
# Performance Issues in Load Distribution

Load distribution performance affects Quality of Service (QoS) perceived by network users. Drawbacks of load distribution models potentially cause poor network performance leading to several problems which are described and discussed as follows.

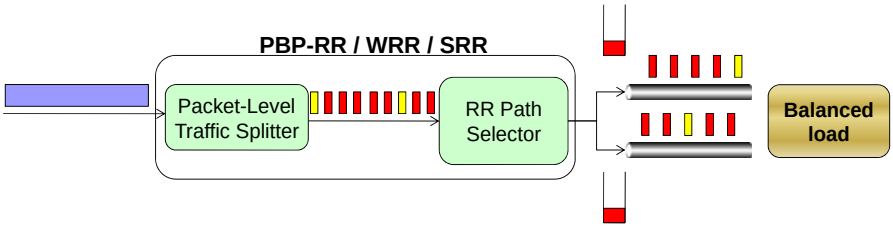
### 3.1. Load Imbalance

Load (i.e., traffic load) is expected to be appropriately shared among multiple paths. Appropriate load sharing can be achieved when the load is assigned on each path properly according to the capacity of the path in terms of, e.g., bandwidth capacity and buffer size. In some specific models, a desired amount of load can be specified as a load threshold. In a path selection, a path being loaded less than the threshold will be chosen for a traffic unit at the arrival instant. If all traffic units have a uniform size, load can be perfectly balanced, i.e., actual load is equal to the desired load. In a practical network, since traffic units (each of which has a random size in the network) inherently take on different sizes, it is difficult to achieve the perfect load balancing. The difference between the desired and actual loads on a particular path is referred to as load balancing deviation. The load imbalance problem occurs when the load balancing deviation exists; that is, the actual load on some path(s) exceeds the desired level while that on some other path(s) falls below. To minimize the load balancing deviation, i.e., for achieving a Coefficient of Variation (CV) of load among the multiple paths that

converges to zero over the large number of traffic units, variance of sizes of one traffic unit must be finite as stated and proved in [42], [45]. These can be depicted by Fig. 3.1.




(a) Load imbalance problem in traditional flow-based load distribution models




(b) Load balancing achieved by packet-based load distribution models

**Figure 3.1.** Examples of performance issue in terms of load balancing efficiency.

Next, we quantitatively describe the load imbalance in terms of the deficit load (of each path) that is a variable representing the gap between the desired and actual loads. By using the boundary condition of a path's deficit load described in [28], the probability of having a certain degree of load imbalance can be roughly quantified as follows. Let $w_p$ be the normalized desired load of path $p$. Let us assume that, over an interval $(0, t]$, $L(t)$ is the load (in packets or in bytes) induced by the $N(t)$ first traffic units. The deficit load of path $p$, referred to as $D_p(t)$, can be calculated as $L_p(t)–w_pL(t)$, where $L_p(t)$ is the actual load of path $p$. The probability of experiencing the deviation

larger than $\xi$ can be expressed, in terms of the average number of traffic units, E[$N(t)$], and CV of the traffic-unit size, $\gamma$, as follows [28]:

$$\Pr[|D_p(t)| > \xi] < \frac{1}{4\xi^2 \, \mathrm{E}[N(t)]}\left(\gamma^2 + 1\right) \tag{3.1}$$

Equation (3.1) shows that the deviation from the desired load depends on $\gamma$ and the number of traffic units, $N(t)$; this equation was proved in [28]. Generally, variation of packet size distribution is bounded by network parameters such as the maximum packet size, whereas that of flow size has no such bound. As compared to flow-level traffic splitting, packet-level traffic splitting has a larger $N(t)$ and a smaller $\gamma$ bounded by certain finite constant of packet size limitation. Having a smaller size of traffic units, load distribution models can achieve more accuracy in load balancing. This is the reason why load distribution models with packet-level traffic splitting can achieve perfect load balance in minimizing load balancing deviation. The load imbalance problem has been studied and several solutions [28], [41], [46] have been proposed. To limit the variance of size of each traffic unit, a traffic unit is split into smaller traffic units; in addition, because of the splitting, the number of traffic units is increased. However, various algorithms have been proposed for making the splitting decision, thus resulting in different improvements and side effects.

Discussions

Since packet-based load distribution models have the smallest traffic unit, with any path selection, they are likely to achieve load balancing as compared to other models with larger traffic unit, according to Equation (3.1). However, this does not

work when the paths have different bandwidth characteristics; PBP-RR can cause load imbalance, i.e., over-utilization on a path with low bandwidth capacity and under-utilization on a path with high bandwidth capacity. WRR, WIRR, SRR, and PWFR can control the amount of load assigned on each path by specifying a weight; they can, with a proper weight, balance load appropriately for each path. EDPF and TS-EDPF can achieve load balancing because of their path selector by using information on network condition; a path having the smallest delay (probably having a small queue length and a low utilization) is selected.

In flow-based models, load imbalance can be attributed to their infinite variation of flow size distribution. The flow-based models equipped with an adaptive algorithm (which can follow dynamic changes in traffic/network conditions) can mitigate the load imbalance problem, by splitting a flow into smaller traffic units, i.e., subflows, in order to reduce variation in the size of traffic units, and by switching a path in order to distribute traffic load. The small traffic unit and intelligent path selector (which accounts for traffic load) are preferred for optimizing load balancing. The following are examples of the flow-based models.

LDM balances load by using an adaptive path selector; a path with a smaller hop-count and lower utilization is more preferred to be selected. In LDM, no path is switched in forwarding a flow; the flow is not split into smaller traffic units. As compared to FS, LDM can achieve better load balancing in normal network operation; however, since there is no splitting, load imbalance can sometimes occur while forwarding a long and high-rate flow of traffic under large variation of flow size

distribution. In AFLCMF, a flow is split when its bit-rate changes such that the flow is classified into a different class. The subflow is sent via a path corresponding to its class. Selecting a path based on bit rate can mitigate the load imbalance problem due to variation of flow size distribution.

LBPF splits only aggressive flows into subflows and moves the subflows to an alternative path which has the shortest queue. Similar to AFLCMF, it can mitigate the load imbalance problem due to variation of flow size distribution. Since it focuses on only the case caused by aggressive flows and ignores that caused by non-aggressive flows, it loses some chance to balance load, and thus cannot achieve perfect load balancing. THR and MBD-/ADBR balance excessive loads of over-utilized paths among under-utilized paths by shifting sub-superflows from over-utilized paths to under-utilized paths. In each control phase, THR moves only one largest sub-superflow while MBD-/ADBR moves several small sub-superflows until all over-utilized paths become under-utilization. Therefore, MBD-/ADBR is likely to achieve better load balancing as compared to THR, which can be confirmed by Equation (3.1). However, THR can also achieve perfect load balance efficiency if its parameters are chosen such that a flow is split into single packets. FLARE splits a flow into subflows and forwards each subflow via a different path which is under-utilized. Similar to THR, FLARE can achieve perfect load balance efficiency if its parameters are chosen such that a flow is split into single packets. However, when packet arrival rate increases, FLARE, splitting only flows having packet inter-arrival time longer than the path difference delay, decreases the number of splits and thus causes load balancing deviation to increase.

In the evaluation as presented in our previous work [71], we directly calculate load balancing deviation in each second from the measured results. Fig. 3.2 illustrates the comparisons in load balancing efficiency of the exemplar models. WRR, which is a packet-based model, can achieve almost perfect load balancing since its load balancing deviation is almost zero, whereas FS and LDM, which are flow-based models, can cause load imbalance since load balancing deviation is very large. LDM having adaptive path selection can reduce load balancing deviation (on average). However, when network utilization increases, the number of packets to be shifted (per time) increases while a path to accommodate the packets tends to have less amount of deficit load; load balancing deviation increases in LDM.

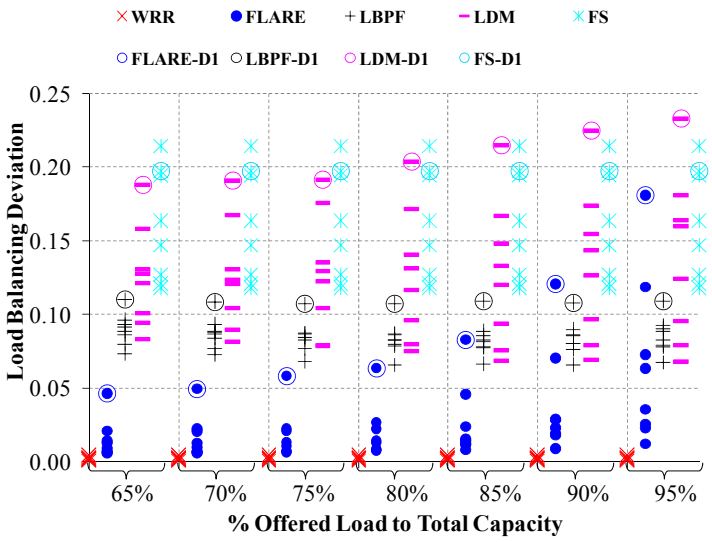



**Figure 3.2.** Comparison in load balancing efficiency.

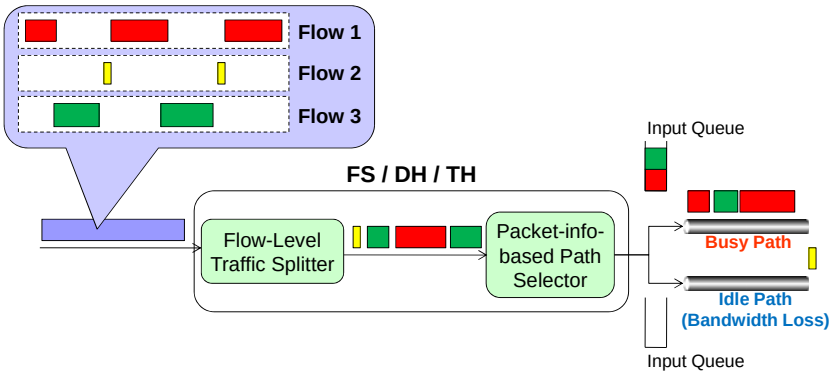
In addition to adaptive path selection, LBPF and FLARE allow splitting of a flow into subflows; load balancing deviation is much smaller. As compared to LBPF'

splitting only aggressive flows, FLARE can further reduce load balancing deviation. When network utilization increases, the splitting rate increases in LBPF but decreases in FLARE. Therefore, load balancing deviation does not increase in LBPF but does increase in FLARE. In Fig. 3.2, the simulation results of trace D1 show that a large variation in flow size distribution causes a large load balancing deviation in each model. (Note that traffic generated from trace D1 has the largest variation in flow size distribution measured in each second. All traces used in [71] obtained from [58].) This observation conforms to the analysis according to Equation (3.1), which describes the relation between load balancing deviation and variation in the size of traffic units. In FLARE, when network utilization is very high and variation in flow size distribution is very large, the splitting rate of FLARE decreases dramatically, thus significantly increasing the load balancing deviation.

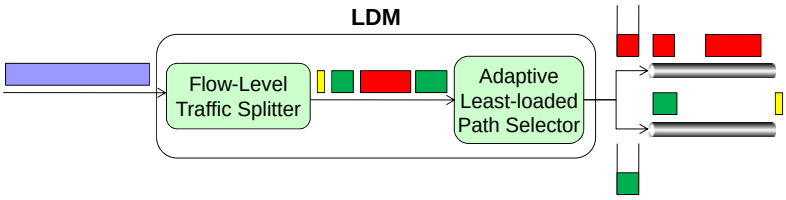
### 3.2. Inefficient Bandwidth Utilization

A load distribution system can be regarded as a work-conserving system (i.e., a system which does not incur waste in utilization of bandwidth resource) if traffic load is perfectly balanced at any time instance such that all outgoing paths are busy or idle at the same time; since no outgoing path is idle while there is input traffic waiting to be forwarded, there is no loss of bandwidth, i.e., efficiency of bandwidth utilization is maximized. Otherwise, it is a non-work-conserving system; at least one path has no load, while the other paths are busy, resulting in bandwidth loss on idle paths. If determination of a path takes into account queue length or level of path utilization, such system can be considered as work-conserving [29]. Otherwise, non-work-conserving

idle time, which is defined as the length of the period when at least one path is idle while others are busy, can increase infinitely if only one particular path is selected for all incoming packets. These can be depicted by Fig. 3.3.




(a) Inefficient bandwidth utilization in traditional flow-based load distribution models




(b) Efficient bandwidth utilization in a flow-based load distribution model with adaptive path selector

**Figure 3.3.** Examples of performance issue in terms of bandwidth utilization efficiency.

Non-work-conservation is affected by the variation in the size of traffic units. If this variation is large, it may cause a long non-work-conservation idle time. Therefore, load distribution models with packet-level traffic splitting and with path selection based on queue length or level of path utilization can achieve the work-conserving property and efficient bandwidth utilization while other models with larger traffic unit and with different path selection schemes deliver less efficient bandwidth utilization.

Discussions

Splitting traffic into single packets causes minimal non-work-conserving idle time while splitting traffic into flows can cause longer non-work-conserving idle time, where the non-work-conserving idle time implies bandwidth loss on idle paths. Packet-based models can achieve a small bandwidth loss whereas flow-based models have a higher loss. Using the RR path selector or selecting the path having the shortest queue, bandwidth loss can be mitigated, and work-conserving property can be achieved. Therefore, packet-based models with the path selectors mentioned above can achieve work-conserving property. However, in WRR and SRR, improper weight assignment can cause non-work-conservation. If a path with low bandwidth is assigned a large weight, a path with large bandwidth assigned a low weight will have an idle period. WIRR implements the interleaving mechanism; the non-work-conserving idle time can thus be reduced.

On performance of flow-based models, variation in flow size distribution, which can be very large, can cause a significant impact. While a particular path is being used to forward a very large flow, other paths (having already finished forwarding shorter flows) are idle resulting in bandwidth loss. Bandwidth utilization efficiency can be affected by a large variation of flow size distribution; in addition, lack of adaptability to current network condition causes this problem to be exacerbated when network utilization increases to the high load condition. In FS, non-work-conserving idle time increases dramatically as the network utilization increases.

In contrast, LDM with adaptability to network conditions selects a least-loaded path; the non-work-conserving time can be decreased. However, since LDM does not allow changing path for a flow, when the network utilization is high, the non-work-conserving idle time is likely to be relatively high, as compared to the other models that allow a flow to be split/re-routed. AFLCMF with adaptability to traffic behavior can switch a large flow to the other path. Similarly, LBPF and FLARE split a flow into several subflows; variation in size of the subflows tends to be smaller. Moreover, a selected path for each subflow can be switched; non-work-conserving idle time can be reduced. In THR and MBD-/ADBR, the selected path is always the most under-utilized path; bandwidth loss can thus be reduced.

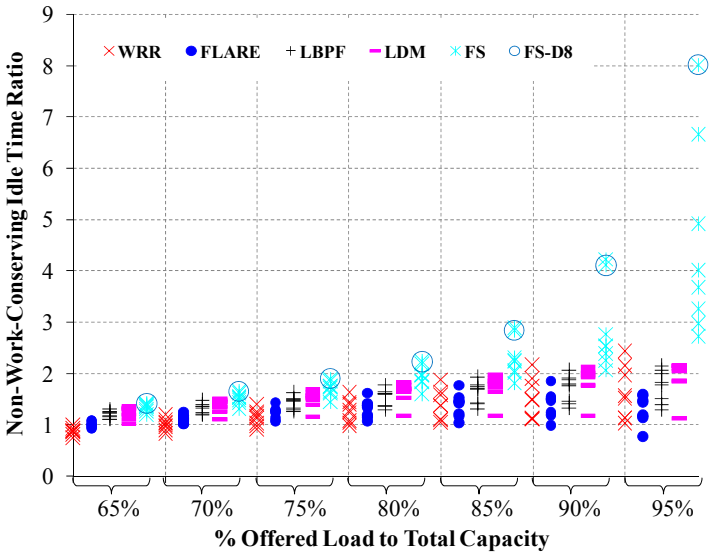



**Figure 3.4.** Comparison in efficiency of bandwidth utilization.

In the evaluation as presented in our previous work [71], we use the non-work-conserving idle time which is the time that all queues are not in the same state (e.g., idle

or busy) to define the metric to evaluate bandwidth utilization efficiency. To compare different models in various conditions, we define the non-work-conserving idle time ratio as the ratio of the accumulated non-work-conserving idle time of all multiple paths to that of the assumed single path having the same aggregated bandwidth. In the best condition, this ratio should be equal to or less than 1. The higher ratio indicates worse bandwidth utilization efficiency because of more bandwidth loss.

As described previously, splitting traffic into single packets can minimize non-work-conserving idle time while splitting traffic into flows can cause longer non-work-conserving idle time, where the non-work-conserving idle time implies bandwidth loss on idle paths. Fig. 3.4 shows that WRR can achieve a small non-work-conserving idle time whereas FS has a longer non-work-conserving idle time. When network utilization increases, non-work-conserving idle time in WRR increases but that in FS increases much more. In FS, the variation in flow size distribution and lack of adaptability to current network conditions dramatically increase the non-work-conserving idle time. The simulation results of trace D8 show that the non-work-conserving idle time can be very long when the variation of flow size distribution is very large. (Note that trace D8 generates traffic having the largest variation measured over all simulation time.) In contrast, LDM with adaptability to network conditions selects the least-loaded path; the non-work-conserving time is thus significantly reduced. In addition to adaptive path selection, LBPF and FLARE allow splitting of a flow into subflows, and thus their non-work-conserving idle time can be further reduced.

### 3.3. Flow Redistribution

Flow redistribution occurs when an original flow is split and re-routed to an alternative path, as shown in Fig. 3.5. The degree of flow redistribution is the number of times that a flow is disrupted by changing the outgoing path for the packets originated from the same flow. For example, it becomes maximized when any successive two packets belonging to the same flow are forwarded via different paths. In a network with multiple paths, changes in the outgoing path can be caused by the increase or decrease in the number of available paths, and the path switching for load balancing. We separately discuss these two factors, i.e., the flow redistribution due to load balancing and the flow redistribution caused by the changes in the number of available paths.

It should be noted that the degree of flow redistribution is totally different from the degree of disruption which is defined as the ratio of the number of flows affected by the increase or decrease in the number of available paths to the total number of flows. The degree of disruption is a performance metric to be used only for flow-based/superflow-based models as mentioned in the previous section.

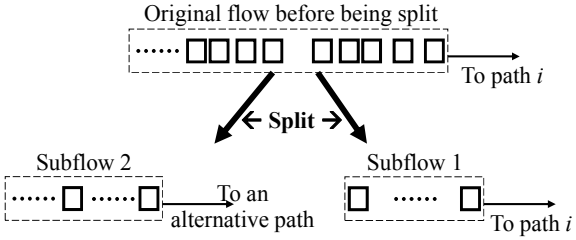



**Figure 3.5.** Flow redistribution.

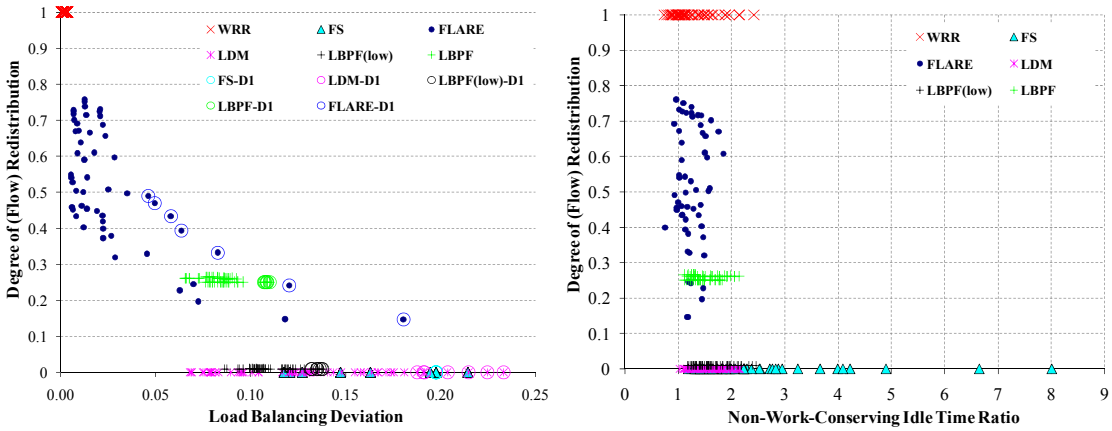
Discussions

When a load balancing mechanism is active, the load adaptation algorithm balances the load between over-utilized paths and under-utilized paths, by moving some flows among the paths, thus causing flows redistribution. In packet-based models, an original flow is split into single packets; the degree of flow redistribution is very high. In flow-based/superflow-based models, flows are in general not split, and thus they do not incur flow redistribution. However, when the number of available paths changes (which is not a normal incident), the splitting of existing flows may become inevitable. This will be described later. The following models allow splitting of a flow, and thus can cause flow redistribution. The degree of flow redistribution depends on the number of affected flows. LBPF may incur only a small degree of flow redistribution because only the aggressive flows are moved. AFLCMF attempts to adjust the flow-rate threshold frequently; a number of flows, which can experience changes of class and path switching, are disrupted. In THR, several flows aggregated in a super-flow are moved. MBD-/ADBR repeatedly moves several super-flows multiple times in each control phase. FLARE redistributes all flows having packet inter-arrival time larger than a certain threshold.

In flow-based/superflow-based models, changes in the number of available paths can cause flow redistribution. In FS, DH, and TH, all flows are re-routed while, in HT, only flows with hash values close to thresholds (i.e., minimum/maximum hash values which are still mapped to the same path) are re-routed. In HRW, PMN-LB, and

LDM, only flows mapped to the deleted/failed path are re-routed; the degree of disruption is very small.

Figs. 3.6(a)–(b) show normalized degrees of flow redistribution, as presented in our previous work [71]. The normalized degree of flow redistribution is quantified by the number of splits divided by the number of successive packets. The maximum value of the normalized degree of splits is 1, in which case input traffic is split into single packets. A value of 0, however, implies no splitting.




(a) Flow redistribution vs. load balancing efficiency (b) Flow redistribution vs. bandwidth utilization efficiency

**Figure 3.6.** Comparison in degree of flow redistribution.

Fig. 3.6(a) illustrates relations between the degree of flow redistribution and load balancing deviation. FS and LDM, which do not split any flow, yield the minimal degrees of flow redistribution at the expense of very large load balancing deviations. In LBPF and FLARE, an increase of the splitting rate causes an increase of the degree of flow redistribution as the price for reducing the load balancing deviation. Since LBPF limits the splitting rate while FLARE does not, LBPF can maintain a smaller degree of

disruption but with a larger load balancing deviation. WRR, which splits a flow into single packets, incurs the maximal degree of flow redistribution but the minimal load balancing deviation. In addition, the simulation results of trace D1 show effects of variation in flow size distribution on the relations between load balancing efficiency and the degree of flow redistribution. LBPF can reduce the load balancing deviation by choosing a higher splitting rate, which causes an increase of degree of flow redistribution. In FLARE, an increase of variation in flow size distribution causes a reduction of the splitting rate, thus resulting in a decrease of the degree of flow redistribution and an increase of the load balancing deviation.

Fig. 3.6(b) depicts relations between the degree of flow redistribution and non-work-conserving idle time. As compared to FS, LDM (which similarly does not cause flow redistribution) yields a smaller non-work-conserving idle time because of its adaptive path-selection. In LBPF, an increase of the splitting rate causes a higher degree of flow redistribution, and can thus reduce the non-work-conserving idle time. FLARE also exhibits similar results. WRR also yields small non-work-conserving idle time. We can see that non-work-conserving idle time can be reduced as the number of splits increases.

### 3.4. Packet Reordering

In the Internet, packet reordering is not a sporadic event [52]. Actually, the packet reordering problem significantly impairs TCP traffic flows (which are mostly found in the Internet) [52], real-time traffic flows, and multimedia traffic flows [53]. In load balancing, packet reordering can occur when the route for a packet of an existing

flow changes; for example, the new route has a lower delay than the old one, as shown in Fig. 3.7.

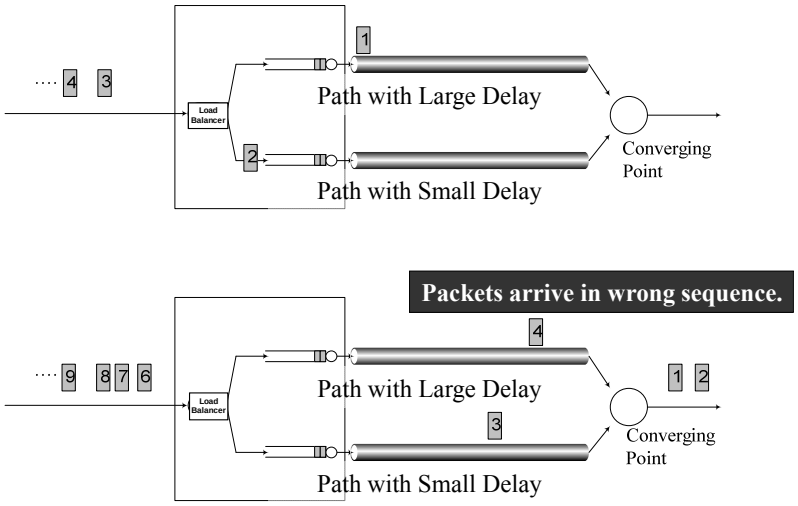



**Figure 3.7.** Occurrence of packet reordering.

Derived in our previous work [69], the risk of packet reordering can be presented in terms of the probability of packet reordering, $\pi_r$, as follows.

$$\pi_r = \pi_s \sum_{i \in \mathbf{P}} \sum_{j \in \mathbf{P}} \Phi(i,j)\Omega(\Delta_{i,j}), \tag{3.2}$$

where $\pi_s$ is the probability of splitting and $\Phi(i,j)$ is the probability of the path switching from path $i$ to path $j$, depending on the path selection strategy; $\Omega(\Delta_{i,j})$ denotes the conditional probability of packet reordering when the path is switched from path $i$ to path $j$, and is a function of $\Delta_{i,j}$, i.e., the difference of end-to-end delays between path $i$ and path $j$. As described in [69], $\Omega(\Delta_{i,j})$ is the cumulative distribution function of the packet inter-arrival time; if $\Delta_{i,j}>0$, $\Omega(\Delta_{i,j})>0$ implies that there is a risk of packet

reordering; otherwise, $\Omega(\Delta_{i,j})=0$, that is, packet reordering will never occur. The smaller value of $\Delta_{i,j}$, the smaller risk of packet reordering. In addition, the occurrence of packet reordering is likely to increase in a network with a number of parallel paths because the probability that packets of a flow take paths with different delays becomes higher [54], [55].

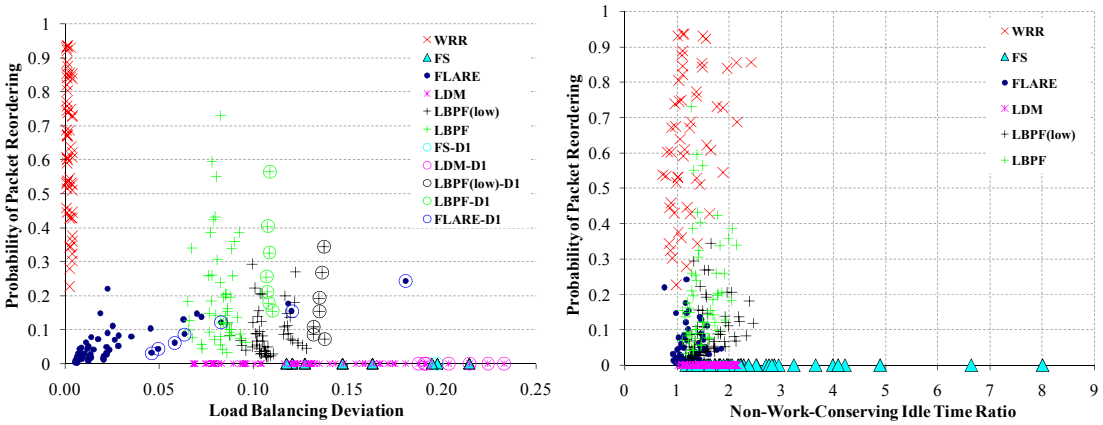
Reordered packets arriving the destination within a certain period of time, referred to as the timeout period, can be successfully recovered via the reordering buffer, at the expense of the increase of packet delay [56], [57]. On the other hand, if reordered packets arrive after the timeout period is over, they are treated as lost packets, thus resulting in not only additional packet delay and but also inefficient network resource utilization for packet retransmissions. In other words, reordering can significantly affect the end-to-end performance as well as network performance. Although it is possible to reduce the occurrence of packet reordering by increasing the size of the reordering buffer, it comes with the price of a longer packet delay. Forwarding all packets bound for the same destination via the same path can completely prevent the reordering problem at the expense of load imbalance. These trade-offs need to be taken into account in mitigating the packet reordering issue.

Discussions

Switching the path of a flow can cause reordering of packets belonging to the flow if the newly selected path has a different delay. All packet-based models, which are non-adaptive models, incur a high risk of packet reordering. WRR and all packet-based models with RR path selection scheme incur a high risk of packet reordering. In

contrast, EDPF and TS-EDPF, selecting the path having the smallest delay, can mitigate the packet reordering problem; however, they are only a little bit better in prevention of packet reordering. Selecting a path based on only the condition of having the smallest delay can also cause a packet to arrive at a destination earlier than a previously sent packet. Without any mechanism to keep the ordering information and to recover the sequence, packet-based models can cause the packet out-of-order problem, thus eventually leading to packet loss. On the other hand, if the required information and packet ordering recovery mechanism are equipped at the destination, packets arrived not in order can be re-sequenced at the expense of an additional delay for waiting for late packets [59], [60]. If the waiting time is too long, the late packets will be treated as packet loss.

Flow-based models send all packets belonging to the same flow via the same path; they can maintain packet ordering. In FS and LDM, there is no risk of packet reordering. With an adaptive load distribution algorithm, the flow can be split and shifted to a different path; such modified flow-based models lose ability to completely prevent packet reordering. AFLCMF and MBD-/ADBR attempt to balance load frequently, and thus they likely cause packet reordering. In contrast, LBPF focusing on minimizing the number of splits can limit the risk of packet reordering; however, the risk of packet reordering is still relatively high as compared to that of FS and LDM. FLARE, splitting a flow conditionally based on traffic and network conditions, can maintain a low risk of packet reordering even under the traffic condition of large variation in flow size distribution.




(a) Packet reordering vs. load imbalance

(b) Packet reordering vs. bandwidth loss

**Figure 3.8.** Performance trade-offs: packet reordering vs. load imbalance and bandwidth loss.

Presented in our previous work [71], Fig. 3.8 illustrates performance trade-offs between load balancing and bandwidth utilization efficiencies, on one hand, and prevention of packet reordering, on the other hand. FS and WRR are two extreme cases where each represents the opposite case. FS, which does not allow splitting of any flow, does not incur any risk of packet reordering whereas load balancing deviation and non-work-conserving idle time are very large. LDM is similar to FS, but it can reduce the non-work-conserving idle time because of its adaptive path selection scheme. LBPF and FLARE, which allow splitting of a flow, incur the risk of packet reordering as the price for reducing the load balancing deviation and non-work-conserving idle time. LBPF with high splitting rate, simply denoted as LBPF, incurs a higher risk of packet reordering but smaller load balancing deviation and non-work-conserving idle time as compared to LBPF with low splitting rate, denoted as LBPF(low). Since FLARE splits only flows which are not expected to incur packet reordering, it can maintain a low risk

of packet reordering while reducing load balancing deviation and non-work-conserving idle time. WRR incurs the minimal load balancing deviation and non-work-conserving idle time, but a very high risk of packet reordering. Simulation results of trace D1 (which generates traffic having the largest flow size variation) show effects of variation in flow size distribution on the trade-off between packet order preservation and load balancing efficiency. As the variation increases, LBPF can mitigate load balancing deviation but cause increased risk of packet reordering. In contrast, FLARE, which avoids splitting a flow having high packet-arrival rate, can maintain a low risk of packet reordering with increased load balancing deviation.

### 3.5. Communication Overhead

To estimate the network condition, some of adaptive load balancing models require communication functions, such as active network probing, network condition gathering, and exchange of network messages, leading to additional traffic which consumes the available bandwidth on the network. The additional traffic does not only decrease the available bandwidth for users, but also increases the network load. Ideally, the communication overhead should be minimized. However, the link state must be updated often enough to minimize the errors in the estimation of network and/or traffic conditions. There is a trade-off between minimizing the communication overhead and improving the load balancing accuracy.

### 3.6. Computational Complexity

In multipath forwarding, a path selection algorithm executed for each packet arrival incurs computational load requiring a processor with enough processing power and resource. Computational complexity is generally used in comparison of various algorithms. A path selection algorithm using constant-sized table has the computational complexity of $O(1)$, whereas the algorithm of finding a path of a list of $n$ paths has the computational complexity of $O(n)$. In this sense, the $O(n)$-complexity algorithm tends to produce a higher processor load. Besides, additional mechanisms for adaptability in selecting path can increase the computational complexity. The overall computational complexity of an adaptive load balancing algorithm with path selection having the computational complexity of $O(g_1(n))$ and the adaptation mechanism having the computational complexity of $O(g_2(n))$ is $O(g_1(n)+g_2(n))$.

### 3.7. Summary

This section describes performance issues in load distribution and then presents performance comparisons among the existing load distribution models which are mentioned in the previous section. Each model (which is described in terms of its internal functions in multipath forwarding mechanism, i.e., the traffic splitting and the path selection) is evaluated by using different criteria, adaptability for dynamic traffic or network condition changes, load balancing and bandwidth utilization efficiencies, packet ordering preservation, degree of flow redistribution, communication overhead, and computational complexity. In our study, it is obvious that the performance of load distribution models largely depends on the feature of their traffic splitting and path

selection schemes. Without the adaptability feature, packet-based models tend to balance load well but cause packet reordering while flow-based models can maintain packet ordering but incur load imbalance. With the adaptability feature, some problems can be solved at the expense of compromising some other advantages.

The comparative performance of existing load distribution models is summarized in Table 3.1. In load balancing efficiency, bandwidth utilization efficiency, and packet order preservation, we represent the degree of the performance by the number of stars from one to three, which can be interpreted as follows. No star, "n/a", means that the problem can occur in normal network operation and can cause severe problem. One star indicates that, only under some specific condition, the problem may not occur. Two stars can be interpreted that the problem may occur (but not frequently), or it can be addressed by some mechanism, or it does not have severe impact on the overall performance. The level of three stars indicates that the problem can be completely prevented or the problem does not cause any significant impact. The special symbol, unshaded star "☆", indicates that the load distribution model can achieve such level under some special condition or with appropriate parameters only.

**Table 3.1.** Comparison of characteristics and performance of load distribution models.

| Model | Traffic splitting level | Path selector | Performance | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Adapt-ability | Load balancing efficiency | Bandwidth utilization efficiency | Packet order preservation | Degree of flow redistri-bution | Degree of dis-ruption | Communi-cation overhead | Computati onal complexity |
| **Info-unaware Models** | | | | | | | | | | |
| PBP-RR [3] | Packet | RR | n/a | ★ | ★★★ | n/a | High | n/a | No | $O(1)$ |
| WRR [33] | Packet | RR,TraffCon (packet counter) | n/a | ★★★ | ★★☆ | n/a | High | n/a | No | $O(1)$ |
| WIRR [35], [36] | Packet | RR,TraffCon (packet counter) | n/a | ★★★ | ★★★ | n/a | High | n/a | No | $O(1)$ |
| SRR [18] | Packet | RR,TraffCon (deficit byte counter) | n/a | ★★★ | ★★☆ | n/a | High | n/a | No | $O(1)$ |
| PWFR [38] | Packet | TraffCon (deficit byte counter) | n/a | ★★★ | ★★☆ | n/a | High | n/a | No | $O(n)$ |
| **Packet-info-based (non-adaptive) Models** | | | | | | | | | | |
| FS [39] | Flow | Packet-Info, RR (for a new flow) | n/a | ★ | ★ | ★★★ | No | High | No | $O(n)$ |
| DH [3] | Flow | Packet-Info | n/a | ★ | ★ | ★★★ | No | High | No | $O(1)$ |
| TH [3] | Super-flow | Packet-Info | n/a | ★ | ★ | ★★★ | No | High | No | $O(1)$ |
| HT [3] | Super-flow | Packet-Info | n/a | ★ | ★ | ★★★ | No | Med-ium | No | $O(1)$ |
| HRW [42] | Flow | Packet-Info | n/a | ★ | ★ | ★★★ | No | Low | No | $O(n)$ |
| PMN-LB [43] | Flow | Packet-Info | n/a | ★ | ★ | ★★★ | No | Low | No | $O(1)$ |
| **Traffic-condition-based Adaptive Models** | | | | | | | | | | |
| AFLCMF [45] | Subflow | Packet-Info, TrafficCon (when traffic condition changes) | Yes | ★★☆ | ★★ | ★★ | Medium | n/a | No | $O(n)$ |
| MBD-/ADBR [47] | Sub-superflow | Packet-Info, TrafficCon (when splitting condition is satisfied) | Yes | ★★☆ | ★★☆ | ★★ | Medium | n/a | No | $O(n)$ |
| **Network-condition-based Adaptive Models** | | | | | | | | | | |
| EDPF [48] | Packet | NetCon | Yes | ★★★ | ★★★ | ★☆ | High | n/a | Yes | $O(n)$ |
| TS-EDPF [49] | Packet | NetCon | Yes | ★★★ | ★★★ | ★☆ | High | n/a | Yes | $O(n)$ |
| LDM [50] | Flow | Packet-Info (for existing flow), NetCon (for a new flow) | Yes | ★☆ | ★☆ | ★★★ | No | Low | Yes | $O(n)$ |
| **Traffic and Network-conditions-based Adaptive Models** | | | | | | | | | | |
| LBPF [41] | Subflow | Packet-Info, TrafficCon when a load adaptation algorithm is activated | Yes | ★☆☆ | ★☆☆ | ★☆☆ | Low-Medium | n/a | No | $O(n)$ |
| | | | | Trade-off * | | | | | | |
| THR [46] | Sub-superflow | Packet-Info, TrafficCon–NetCon when splitting condition is satisfied) | Yes | ★★☆ | ★★☆ | ★★☆ | Medium-High | n/a | No | $O(n)$ |
| | | | | Trade-off * | | | | | | |
| FLARE [28] | Subflow | Packet-Info, TrafficCon when delay-based splitting condition is satisfied | Yes | ★★☆ | ★★☆ | ★★★ | Medium-High | n/a | Yes | $O(n)$ |
| | | | | Trade-off ** | | | | | | |

★: Only under some specific condition, the problem may not occur.
★★: Problem may occur, but not frequently or can be addressed by some mechanism or does not have severe impact on overall performance.
★★★: Problem can be completely prevented or the problem does not cause any significant impact.
☆: Such level can be achieved under some special condition or with appropriate parameters only.
* One side is load balancing and bandwidth utilization; the other side is packet order preservation and degree of flow redistribution.
** One side is load balancing and bandwidth utilization; the other side is degree of flow redistribution.

# Chapter 4

# Effective Delay-Controlled Load Distribution

## 4.1. Problems and Motivations

Load distribution models have been applied in various kinds of networks and for a variety of service applications as mentioned in the introduction, and research on load distribution algorithms has been studied for many years. However, most of the researches do not focus directly on latency which has a significant impact on QoS required for multimedia and real-time applications. The demand for network infrastructure in providing low latency and low variation of latency network services that can support the delay-sensitive applications is a major driving force for this work.

Delay-related Issues

Total packet-delay is the time to successfully transmit a packet, i.e., end-to-end delay in transmitting a packet and additional time required in packet reordering recovery. 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 fixed delay (i.e., propagation delay), $D_p$, and queueing delay, $Q_p$. 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 ($D_r$), i.e., packet delay = $D_p + Q_p + D_r$, whereas "packet delay variation" refers to the variation in the end-to-end delay of packets successively arrived at a destination. 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 $D_r$. 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 [57], [72], [73].

Flow-based models can completely prevent packet reordering. The major drawback of the flow-based models is the inability to deal with variation of flow size distribution [41], thus leading to the load imbalance problem. Flow-based models can cause large variation in packet delay, affected from overload and, consequently, the large $Q_p$ (causing a large end-to-end delay) on a particular path. Variants of flow-based models, e.g., LBPF and FLARE, allow switching a path for some of the packets in the same flow (increasing $\pi_s$ and $\Phi(i,j)$ in Equation (3.2)) improve load balancing efficiency at the price of a risk of packet reordering (increasing $\pi_r$), and vice versa. So there is a trade-off between small $Q_p$ and small $D_r$. Therefore, packet delay cannot be effectively reduced by the existing load distribution models.

Possible Solution

Since packet-based load distribution models having a large $\pi_s$ and $\Phi(i,j)$ can achieve competent load balancing efficiency, they can minimize $Q_p$. However, the major drawback is their inability to maintain per-flow packet ordering. This leads to a high degree of packet reordering [54], [55], [56], thus resulting in the large $D_r$ induced by packet reordering recovery. According to Equation (3.2), probability of packet reordering can decrease when $\Delta_{i,j}$ decreases, as illustrated by Fig. 4.1. Therefore, a load

distribution model that can effectively reduce total packet delay should be a packet-based model which can reduce $\Delta_{i,j}$ without decreasing $\pi_s$ and $\Phi(i,j)$.

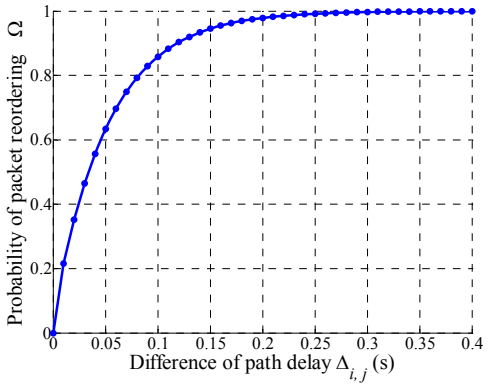



**Figure 4.1.** Probability of packet reordering when path is switched.

## 4.2. Model Descriptions

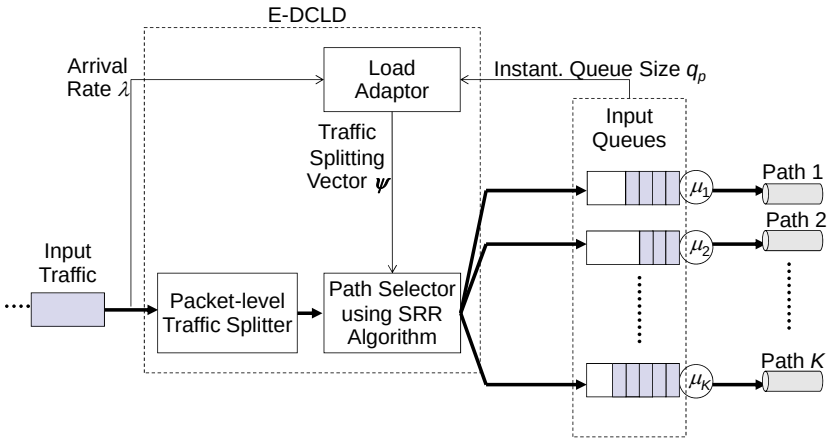



**Figure 4.2.** Description of the proposed model, E-DCLD.

We propose Effective Delay-Controlled Load Distribution (E-DCLD) model [70] that can outperform the existing models in solving the delay-related problems. Fig. 4.2 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 in load distribution control, and thereby properly responding to network condition without additional network overhead. In the path selector, we implement the surplus-round-robin (SRR) load sharing algorithm [18] 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 our previous work, DCLD [68], are detailed in the next subsection.

### 4.3. Load Distribution Control

Let **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,

$$C_p(\psi_p) = D_p + (1-w)\frac{1}{\mu_p - \psi_p \lambda} + w\frac{q_p}{\mu_p} \quad (4.1)$$

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 queueing delay which varies according to the input traffic rate ($\lambda$), the bandwidth capacity of the path ($\mu_p$), and the traffic splitting ratio ($\psi_p$). With the assumption that input traffic is a combination of Poisson traffic and unknown traffic which cannot be identified, the queueing delay is modeled as a mixture of an M/M/1 queue (which has low complexity as compared to other queueing models) and a measurement. Therefore, with a weight factor $w$, the queueing delay is obtained by averaging the second term which is the average queueing 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 queueing model, thus measured as $q_p/\mu_p$. With a small value, $w \rightarrow 0$, E-DCLD calculates the queueing 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 queueing delay is calculated only from the queue size, which is almost similar to the LLF model (i.e., a packet-based model with LLF path selection scheme) that can decrease the average queue size but is likely to increase the risk of packet reordering. From Equation (4.1), the optimal splitting vector can be derived by solving the optimization problem:

$$\text{Maximize} \quad \max_{p \in \mathbf{P}} C_p(\psi_p), \tag{4.2}$$

$$\text{Subject to} \quad \sum_{p \in \mathbf{P}} \psi_p = 1 \quad \text{and} \quad 0 \le \psi_p \le \frac{\mu_p}{\lambda} \le 1.$$

The traffic splitting vector, $\boldsymbol{\psi}^{\boldsymbol{n}} = \{\psi_{\boldsymbol{p}}^{\boldsymbol{n}}\}$ for all $p \in \mathbf{P}$, consists of the control variables of the problem described in Equation (4.2) and the proportion of traffic allocated to path $p$ at time $t_n$. The initial splitting vector, $\boldsymbol{\psi}^0$, is calculated from Equation (4.3).

$$\forall p \in \mathrm{P} : \psi_p^0 = \frac{\mu_p}{\sum_{p \in \mathbf{P}} \mu_p} \tag{4.3}$$

**4.4. Load Adaptation Algorithm**

When the $m^{\text{th}}$ packet arrives (at a diverging point of input traffic), the packet arrival rate $\lambda$ 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 Equation (4.1). 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. For each arrived packet, the load adaptor performs the load adaptation algorithm (to adjust traffic splitting vector) described in Fig. 4.3 and change of path costs can be shown in Fig. 4.4.

1. Calculate $C_p(\psi_p)$ by using Equation (4.1) 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

$$C_{p_{worst}}\left(\psi_{p_{worst}} - \Delta\psi\right) = C_{p_{best}}\left(\psi_{p_{best}} + \Delta\psi\right) \tag{4.4}$$

where

$$\Delta\psi = \begin{cases} \dfrac{S_\Delta}{2\lambda} & ;\Delta D + w\Delta K = 0 \\ \dfrac{S_\Delta + \dfrac{2(1-w)}{\Delta D + w\Delta K} - \upsilon\sqrt{(S_\Sigma)^2 + \left(\dfrac{2(1-w)}{\Delta D + w\Delta K}\right)^2}}{2\lambda} & ;\Delta D + w\Delta K \neq 0 \end{cases}$$

$S_\Delta = (\mu_{p_{best}} - \lambda\psi_{p_{best}}) - (\mu_{p_{worst}} - \lambda\psi_{p_{worst}})$, $S_\Sigma = (\mu_{p_{best}} - \lambda\psi_{p_{best}}) + (\mu_{p_{worst}} - \lambda\psi_{p_{worst}})$, $\Delta D = D_{p_{best}} - D_{p_{worst}}$, $\Delta K = \dfrac{q_{p_{best}}}{\mu_{p_{best}}} - \dfrac{q_{p_{worst}}}{\mu_{p_{worst}}}$, and $\upsilon = \dfrac{|\Delta D + w\Delta K|}{\Delta D + w\Delta K}$.

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

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

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

**Figure 4.3.** Load adaptation algorithm for E-DCLD.

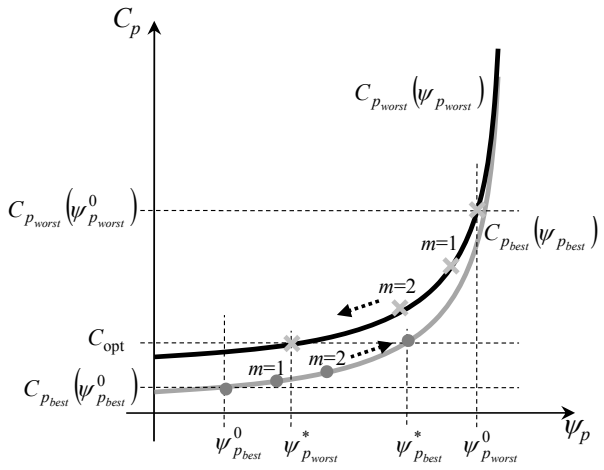



**Figure 4.4.** Change of path costs.

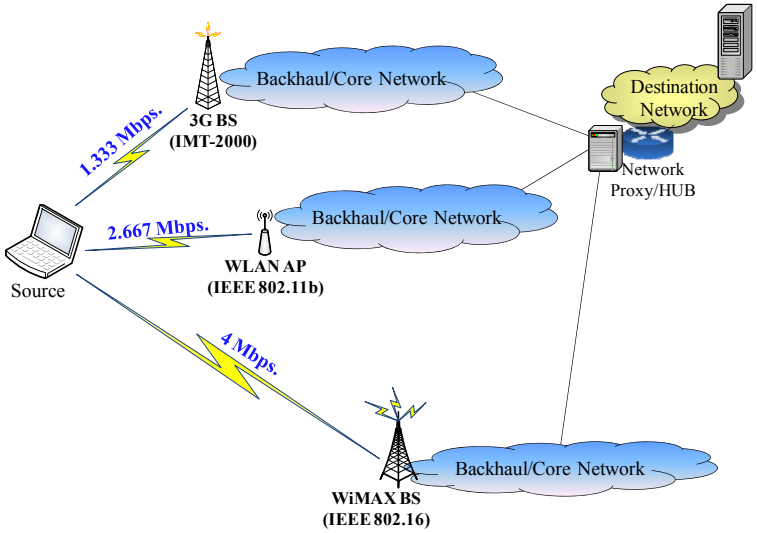
For each packet arrival, $m$, the splitting vector is adjusted and the difference among the path costs is reduced, according to Equation (4.4). When $m \to \infty$, the cost of each path will converge to the same value, which allows us to achieve the objective function in Equation (4.2). The proofs of convergence and optimization are given in [70].

## 4.5. Performance Analysis

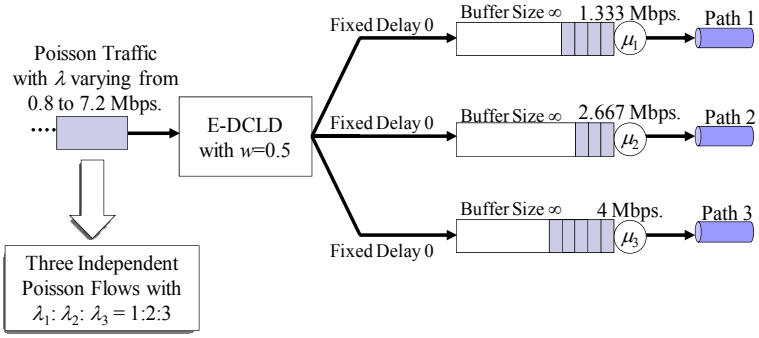
### 4.5.1. Simulation Environment

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. 4.5 from the view point of a source having

multiple paths to a destination. Fig. 4.5(a) demonstrates the multiple paths established over 3G network, wireless LAN, and WiMAX. Fig. 4.5(b) shows an analytical model of the multipath network.




(a) Network topology




(b) Analytical model of the multipath network

**Figure 4.5.** Simulation environment – Poisson input traffic.

The input traffic from the source will be split into three multiple paths ($K$=3) having aggregated bandwidth ($\mu$) 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 ($\lambda/\mu$) 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 run-time 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 1000, and period of adaptation ($P$) is 20; that is, the table will be updated for every 1000 packets and the largest flow recorded in the table will be switched to a new path for every 20 packets.

### 4.5.2. End-to-End Delay

This is an important network latency experienced by all packets and constituted by propagation delay and queueing delay. Theoretically, if the input traffic is Poisson and path $p$ is randomly selected with probability $\psi_p$ 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 $q_p/\mu_p$ is the (expected) waiting time of the packet in the queue, the cost value obtained from the cost function $C_p$ in Equation (4.1) will be close to the (expected) end-to-end delay of path $p$. 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 $\boldsymbol{\psi}^*$, all paths have (almost) the same delay. The maximum path delay is minimized and the end-to-end delay is therefore reduced.

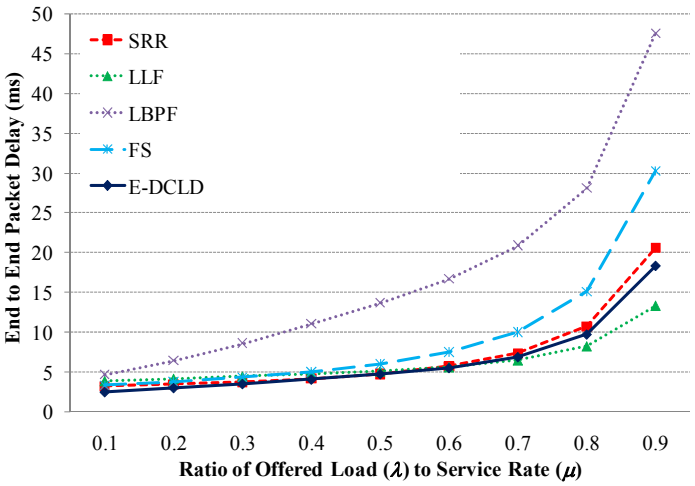



**Figure 4.6.** Mean end-to-end delay when input traffic is Poisson.

Fig. 4.6 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 [18]) 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. 4.6 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 compatible with a real network, implying that its end-to-end delay is likely to be much larger than that in the simulation.

### 4.5.3. Packet Delay Variation

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 and 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. 4.7 presents the coefficient of variation (CV) among end-to-end delays of all candidates.

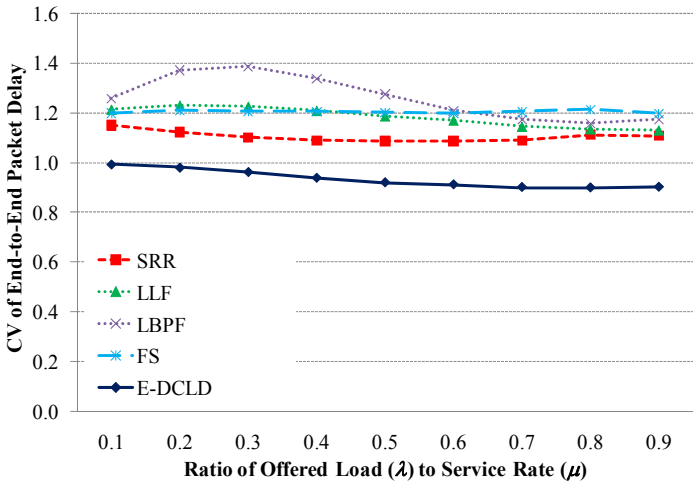



**Figure 4.7.** Coefficient of variation of end-to-end delay when input traffic is Poisson.

Fig. 4.8 shows that 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.

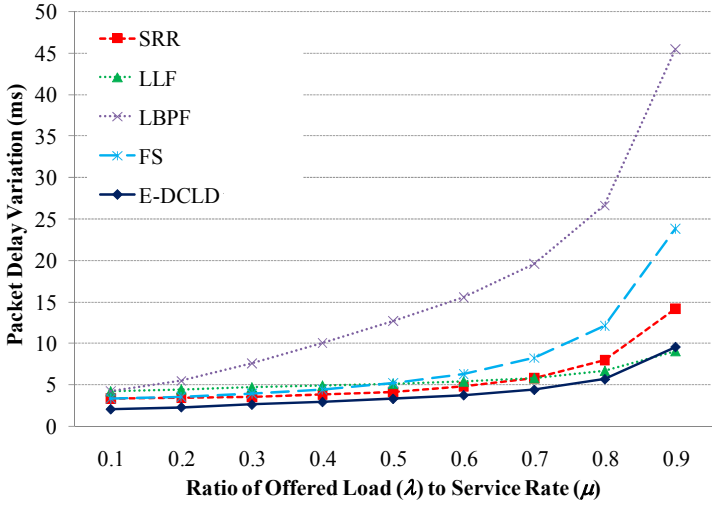



**Figure 4.8.** Packet delay variation when input traffic is Poisson.

### 4.5.4. Risk of Packet Reordering

According to Equation (3.2), E-DCLD aiming to minimize $|\Delta_{i,j}|$ strives to maintain a low risk of packet reordering [69], [70]. As compared to E-DCLD, packet-based models such as SRR and LLF can cause a high risk of packet reordering [67]. Especially, LLF, which only chooses the path with the shortest queue, is highly likely to have $\Delta_{i,j}>0$, implying that it can cause a high risk of packet reordering. Fig. 4.9 shows that E-DCLD, which can decrease the variation among end-to-end delays as illustrated in Fig. 4.7, 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.

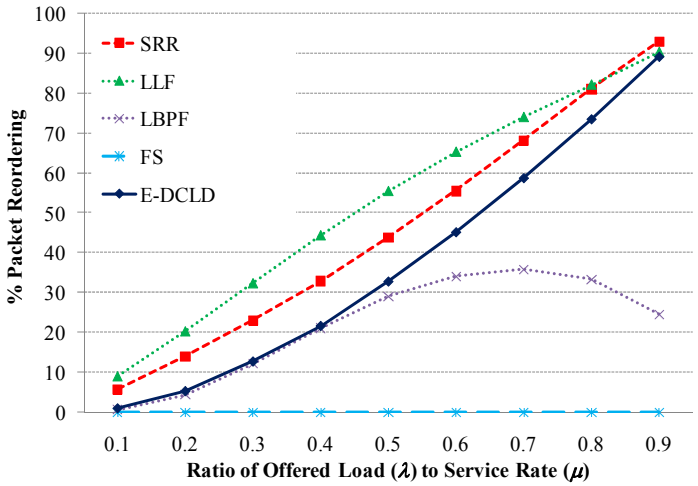



**Figure 4.9.** Risk of packet reordering when input traffic is Poisson.

### 4.5.5. 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. 4.10, E-DCLD achieves both low end-to-end delay and low risk of packet reordering, and thus can maintain a small (total) packet delay.

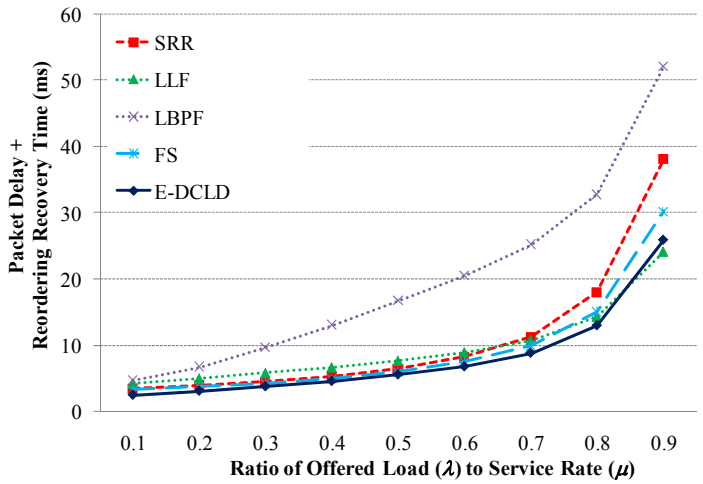
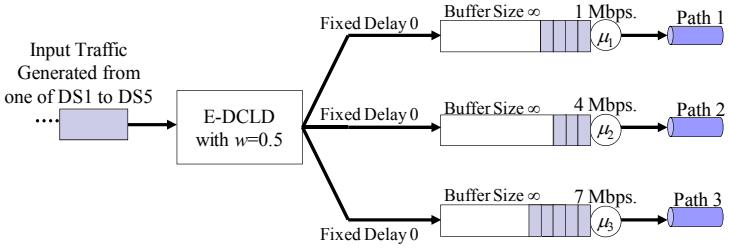


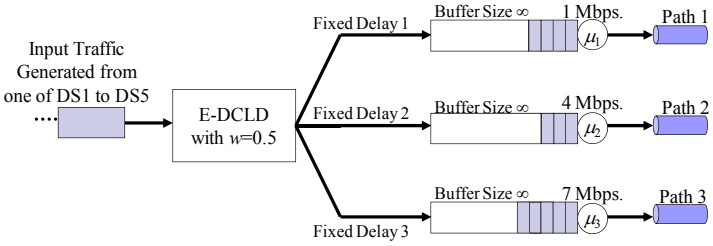

**Figure 4.10.** Mean total (packet) delay when input traffic is Poisson.

## 4.6. Performance Evaluations Based on Real Traffic

### 4.6.1. Simulation Environment




(a) Simulation scenario I – Equal fixed delays: $D_1$=$D_2$=$D_3$=0




(b) Simulation scenario II – Unequal fixed delays: $D_1$=1 ms, $D_2$=2 ms, $D_3$=3ms

**Figure 4.11.** Simulation scenarios – input traffic generated from traces of real traffic.

We demonstrate and discuss comparative performance under various conditions of real traffics (not Poisson) under the environment shown in Fig. 4.11. Simulations in Fig. 4.11(a) 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 Fig. 4.11(b) are conducted to demonstrate the superior performance of E-DCLD in such a realistic environment. This will be discussed in the next subsections. Simulation setups in this subsection are almost similar to that in the previous subsection with the following exceptions.

For each simulation scenario, five simulation sub-scenarios are conducted to show the performance of each load distribution model, by using 1-hour long real traffic traces [58], 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 4.1 and depicted by Fig. 4.12. 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.

**Table 4.1.** Profile of traffic traces.

| Trace ID | # Packets x$10^6$ | 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 |

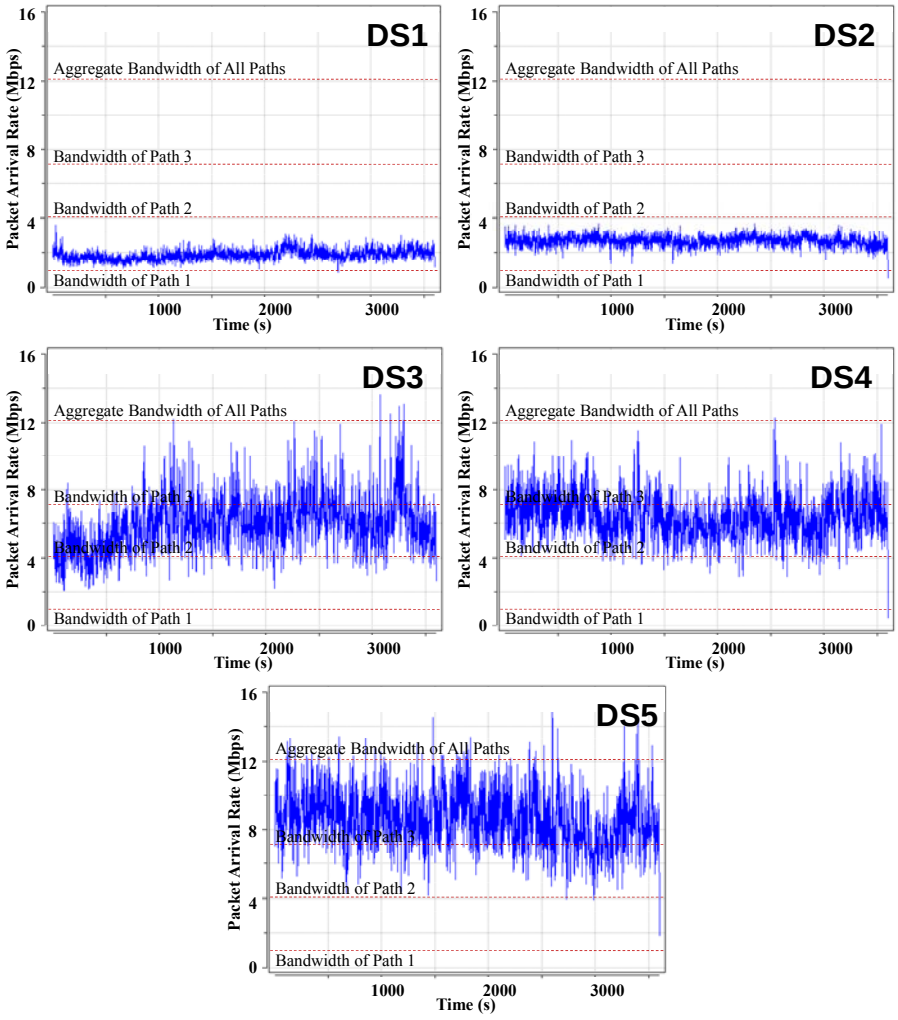



**Figure 4.12.** Traffics characteristics.

With the set-up 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, $W$=1000, and $P$=20. In FLARE, $\delta$ is set to 50 ms (i.e., minimum of inter-arrival 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.

### 4.6.2. Simulation Scenario I – Equal Fixed Delays

In this simulation scenario, all fixed delays are assumed to be equal: $D_1 = D_2 = D_3 = 0$. Performance comparisons are presented and discussed as follows.

End-to-End Delay

Fig. 4.13 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, when traffic load is so low that two (or more) queues are idle, LLF cannot

find the smallest-delay path. 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 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.

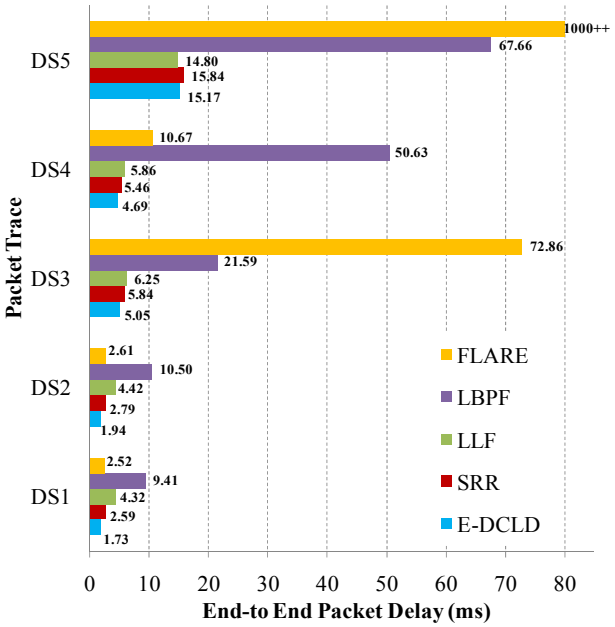



**Figure 4.13.** Mean end-to-end delay under input traffic generated from traces of real traffic and multiple paths having $D_1=D_2=D_3=0$.

Packet Delay Variation

Fig. 4.14 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. 4.15 shows that E-DCLD can efficiently mitigate variation in the end-to-end delay caused by the overloaded paths. Fig. 4.15(a) 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. 4.15(b)–(f) demonstrate the performance among all models, and the evidence that E-DCLD can maintain the smallest delay variation. Under various traffic conditions, Fig. 4.16 shows packet delay variations achieved by various models, and thus clearly demonstrates the superiority of E-DCLD.

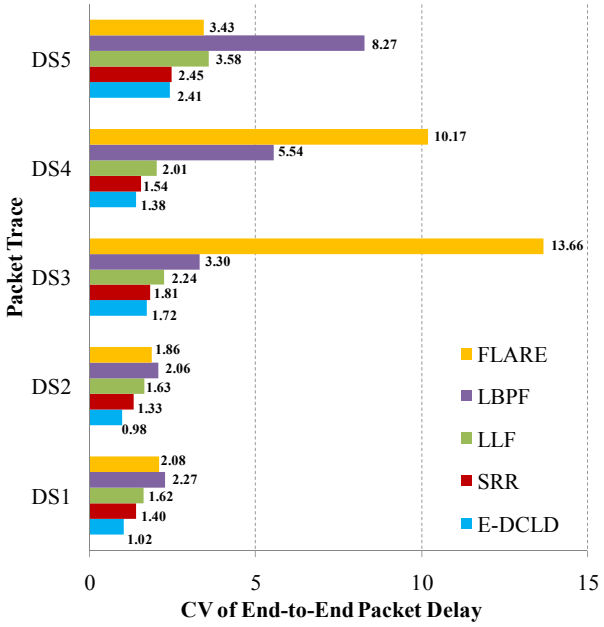



**Figure 4.14.** Coefficient of variation of end-to-end delay under input traffic generated from traces of real traffic and multiple paths having $D_1=D_2=D_3=0$.

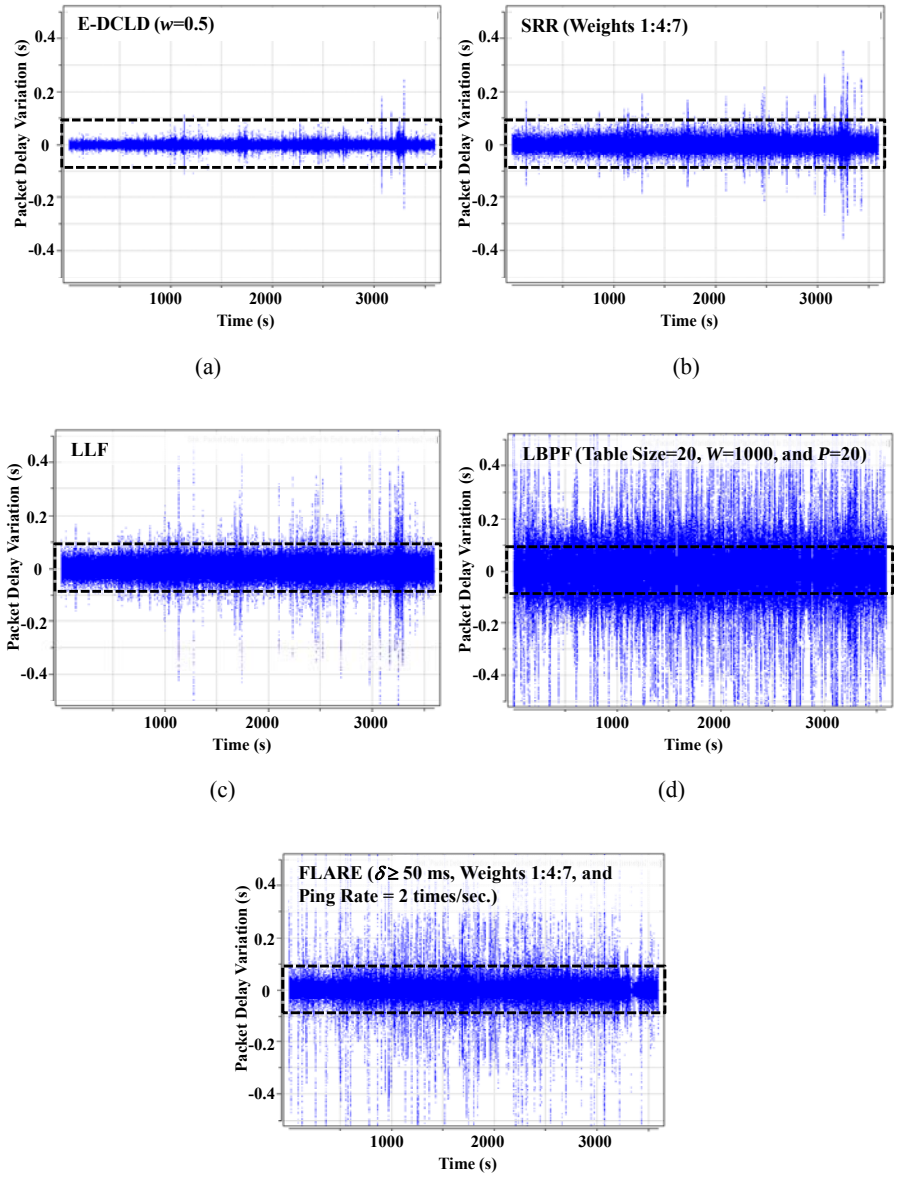



**Figure 4.15. (a)–(e)** Packet delay variation under traffic generated from trace DS3 when load distribution models, E-DCLD, SRR, LLF, LBPF, and FLARE, are employed, respectively, and multiple paths having $D_1$=$D_2$=$D_3$=0.

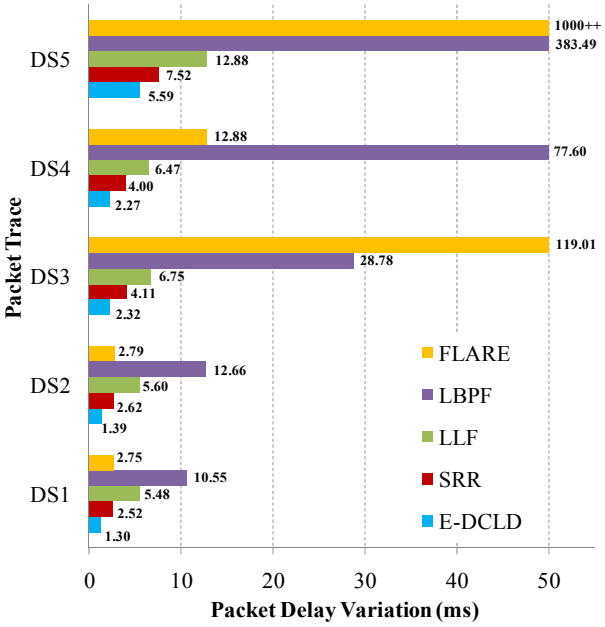



**Figure 4.16.** Packet delay variation under input traffic generated from traces of real traffic and multiple paths having $D_1=D_2=D_3=0$.

Risk of Packet Reordering

Fig. 4.17 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 parameters. LBPF splits a group of largest flows, thus causing the risk of packet reordering. FLARE splits only flows with packet inter-arrival time which is small enough, and hence does not cause packet reordering [39], [41], thus minimizing the risk of packet reordering.

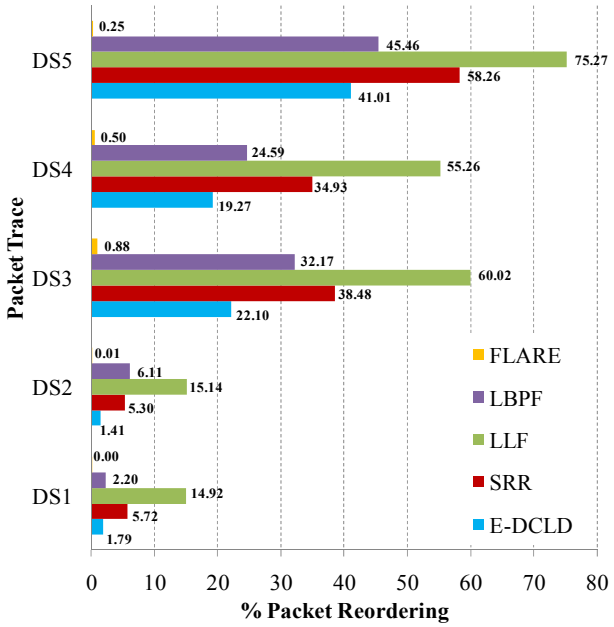



**Figure 4.17.** Risk of packet reordering under input traffic generated from traces of real traffic and multiple path having $D_1=D_2=D_3=0$.

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. 4.18. 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.

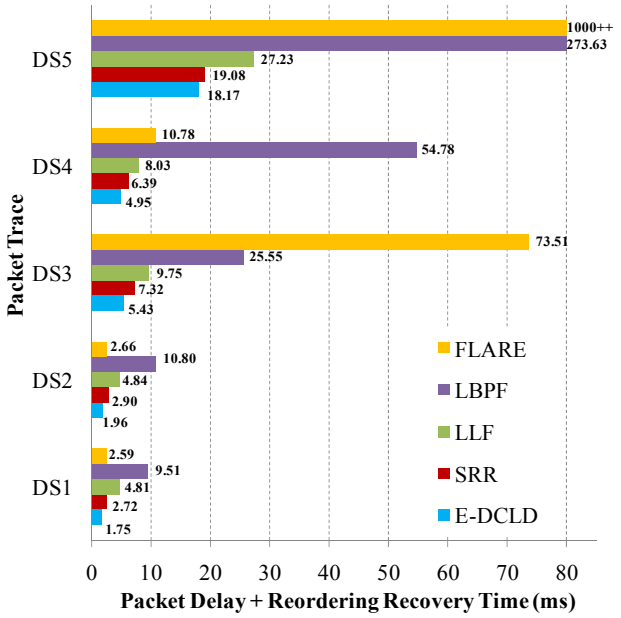



**Figure 4.18.** Mean (total) packet delay under input traffic generated from traces of real traffic and multiple path having $D_1$=$D_2$=$D_3$=0.

### 4.6.3. Simulation Scenario II – Unequal Fixed Delays

When each path has different fixed delays: $D_1$ = 1 ms, $D_2$ = 2 ms, and $D_3$ = 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 4.2 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, which depicts effect of fixed delay on load distribution control.

**Table 4.2.** Ratio of the number of packets sent via each path.

| 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 |

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. 4.19) and variation among the end-to-end delays (as illustrated in Fig. 4.20) such that the packet delay variation and risk of packet reordering can be significantly reduced, as illustrated in Fig. 4.21 and Fig. 4.22, respectively. Likewise, the packet delay can be decreased as illustrated in Fig. 4.23.

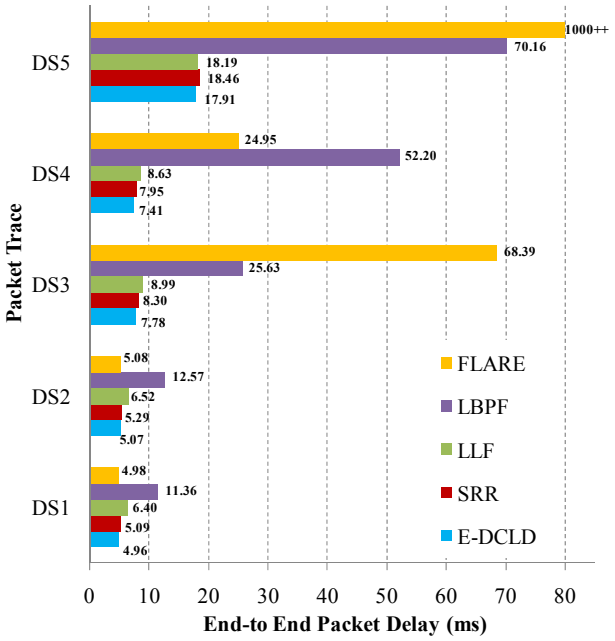



**Figure 4.19.** Mean end-to-end delay under input traffic generated from traces of real traffic and multiple paths having $D_1$=1, $D_2$=2, $D_3$=3.

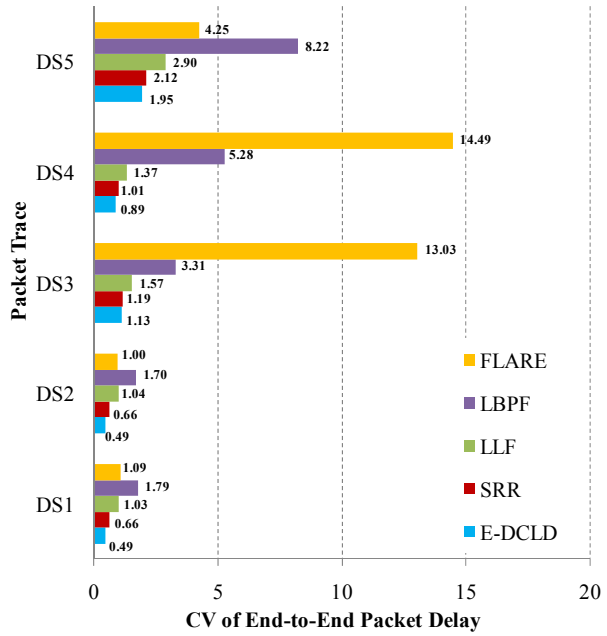



**Figure 4.20.** Coefficient of variation of end-to-end delay under input traffic generated from traces of real traffic and multiple paths having $D_1$=1, $D_2$=2, $D_3$=3.

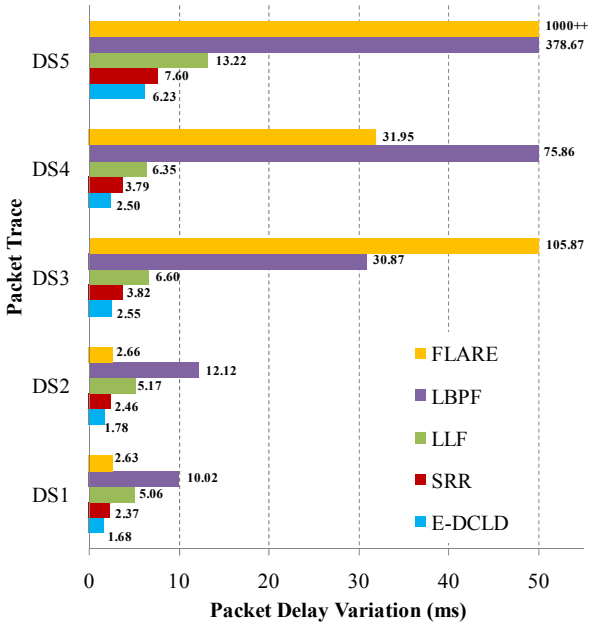



**Figure 4.21.** Packet delay variation under input traffic generated from traces of real traffic and multiple paths having $D_1$=1, $D_2$=2, $D_3$=3.

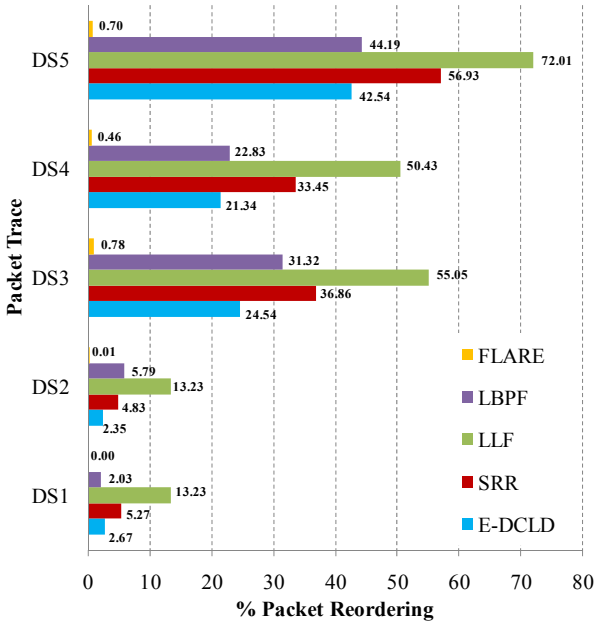



**Figure 4.22.** Risk of packet reordering under input traffic generated from traces of real traffic and multiple paths having $D_1$=1, $D_2$=2, $D_3$=3.

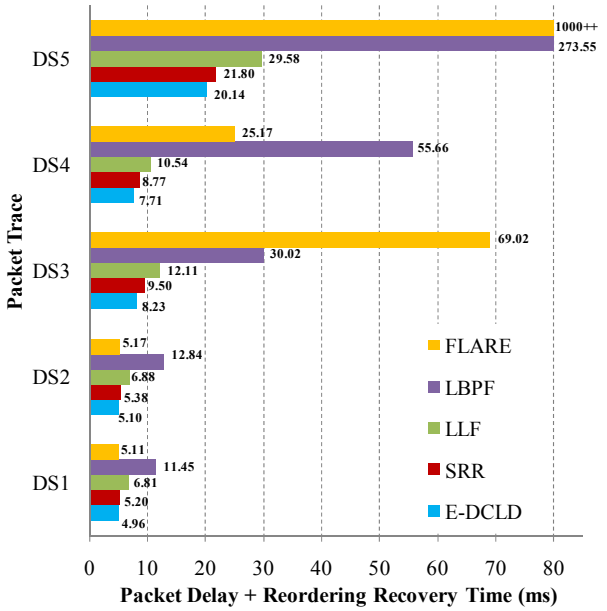



**Figure 4.23.** Mean (total) packet delay under input traffic generated from traces of real traffic and multiple paths having $D_1$=1, $D_2$=2, $D_3$=3.

## 4.7. Summary

Since an effective model of load distribution is important 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, 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.

# Chapter 5

# Concluding Remarks

The demand for network infrastructure in providing high-speed and high-quality broadband network services that can support multimedia and real-time applications has been motivation for research and development of a traffic load distribution scheme. Bandwidth aggregation and network-load balancing are challenging research problems, and a large number of traffic load distribution approaches have been proposed. However, a majority of the solutions do not focus on delay-related issues which have a significant impact on multimedia and real-time applications. The primary contributions of this dissertation are a survey on load distribution models and a study towards an effective model of load distribution for the delay-sensitive applications.

The first primary research contribution is a comprehensive review of various load distribution models, which provides useful information for research in this area, e.g., collection, summarized descriptions, classification, analysis, and comparison of the existing load distribution models. Each model is described and classified in terms of its internal functions in multipath forwarding mechanism, i.e., the traffic splitting and the path selection. Significant performance issues in load distribution are presented. The performance of each model is evaluated by using different criteria, adaptability for dynamic traffic or network condition changes, load balancing and bandwidth utilization efficiencies, packet ordering preservation, degree of flow redistribution, communication overhead, and computational complexity. In this study, it is obvious

that the performance of load distribution models largely depends on the feature of their traffic splitting and path selection schemes. Moreover, we proposed an analytical model which can be used to estimate risk of packet ordering.

In the second primary research contribution, based on the study of existing load distribution models obtained from the survey, we proposed an effective model of load distribution (i.e., E-DCLD) that is essential to efficiently utilize multiple parallel paths for multimedia data transmission and real-time applications. First, delay-related issues caused by load distribution, which is necessary for developing a delay-controlled load distribution model, are described. Then we present E-DCLD which aims to minimize the difference among end-to-end delays by using locally available information. Variation in end-to-end delay can be reduced and thus the packet delay variation and risk of packet reordering are 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, E-DCLD can overcome delay-related issues. 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.

# Bibliography


[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, Sep. 2002.

[2] R. Martin, M. Menth, and M. Hemmkeppler, "Accuracy and dynamics of multi-stage load balancing for multipath Internet routing," in *Proc. IEEE ICC*, Glasgow, Scotland, Jun. 2007, pp. 6311–6318.

[3] C. Villamizar, "OSPF optimized multipath (OSPF-OMP)," Internet draft draft-ietf-ospf-omp-02.txt, Feb. 1999.

[4] D. Thaler and C. Hopps, "Multipath issues in unicast and multicast next-hop selection," RFC 2991, Nov. 2000.

[5] J. Moy, "OSPF version 2," RFC 2328, Apr. 1998.

[6] G. Malkin, "RIP version 2," RFC 2453, Nov. 1998.

[7] J. Kulkarni and N. Anand, "Equal cost routes support for RIP/RIPNG," draft-janardhan-naveen-rtgwg-equalcostroutes-rip-00, Jun. 2007.

[8] (Online Sources) Cisco Systems Inc., "Enhanced interior gateway routing protocol (EIGRP)," Cisco white paper EIGRP, Available: http://www.cisco.com/warp/public/103/eigrp-toc.html or http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a0080094cb7.shtml.

[9] E. Rosen, A. Viswanathan, and R. Callon "Multiprotocol label switching architecture," RFC 3031, Jan. 2001.

[10] B. Jamoussi, L. Andersson, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis, "Constraint-based LSP setup using LDP," RFC 3212, Jan. 2002.

[11] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP tunnels," RFC 3209, Dec. 2001.

[12] Y. Wang and Z. Wang, "Explicit routing algorithms for Internet traffic engineering," in *Proc. International Conference on Computer Communication Networks (ICCCN'99)*, Boston, MA, Sep. 1999, pp. 582-588.

[13] R. Roy and B. Mukherjee, "Degraded-Service-Aware multipath provisioning in telecom mesh networks," in *Proc. IEEE/OSA Optical Fiber Communications (IEEE/OSA OFC 2008)*, San Diego, CA, Feb. 2008.

[14] M. Menth, R. Martin, A. Koster, and S. Orlowski, "Overview of resilience mechanisms based on multipath structures," in *Proc. the 6th International Workshop on Design and Reliable Communication Networks (DRCN)*, La Rochelle, France, Oct. 2007.

[15] J. Duncanson, and A. Berger, "Inverse multiplexing," IEEE Communications Magazine, vol. 32, pp. 34–41, Apr. 1994.

[16] P. H. Fredette, "The past, present, and future of inverse multiplexing," IEEE Communications Magazine, vol. 32, pp. 42–46, Apr. 1994.

[17] A. C. Snoeren, "Adaptive inverse multiplexing for wide area wireless networks," in *Proc. IEEE GLOBECOM*, Rio de Janeiro, Brazil, Dec. 1999, pp. 1665–1672.

[18] H. Adiseshu, G. Parulkar, and G. Varghese, “A reliable and scalable striping protocol,” ACM SIGCOMM Computer Communication Review, vol. 26, no. 4, pp. 131–141, Oct. 1996.

[19] J.-Y. Jo, Y. Kim, H. J. Chao, and F. Merat, “Internet traffic load balancing using dynamic hashing with flow volume,” in *Proc. SPIE ITCom 2002*, Boston, MA, Aug. 2002.

[20] S. J. Lee and M. Gerla, “Split multipath routing with maximally disjoint paths in ad hoc networks,” in *Proc. IEEE ICC*, Helsinki, Finland, Jun. 2001, pp. 3201–3205.

[21] L. Wang, Y. Shu, M. Dong, L. Zhang, and O. Yang, “Adaptive multipath source routing in ad hoc networks,” in *Proc. IEEE ICC*, Helsinki, Finland, Jun. 2001, pp. 867–871.

[22] D. Johnson, Y. Hu, and D. Maltz, “The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4,” RFC 4728, Feb. 2007.

[23] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, “A framework for reliable routing in mobile ad hoc networks,” in *Proc. IEEE INFOCOM*, CA, Mar. 2003, pp. 270–280.

[24] M. K. Marina and S. R. Das, “Ad hoc on-demand multipath distance vector routing,” Wireless Communications and Mobile Computing, vol. 6, no. 7, pp. 969–988, Nov. 2006.

[25] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing,” RFC 3561, Jul. 2003.

[26] T. Taleb, D. Mashimo, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "Explicit load balancing technique for NGEO satellite IP networks with on-board processing capabilities," IEEE/ACM Trans. Networking, vol. 17, no. 1, pp. 281–293, Feb. 2009.

[27] J. Postel, "Internet protocol: DARPA Internet program protocol specification," RFC 791, Sep. 1981.

[28] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic load balancing without packet reordering," ACM SIGCOMM Computer Communication Review, vol. 37, no. 2, pp. 53–62, Apr. 2007.

[29] Z. Cao, Z. Wang, and E. Zegura, "Performance of hashing based schemes for Internet load balancing," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 332–341.

[30] (Online Sources) Cisco Systems Inc., *DNS Server Round-Robin Functionality for Cisco AS5800*, Available: http://www.cisco.com/en/US/docs/ios/12_1t/12_1t3/feature/guide/dt_dnsrr.html

[31] (Online Sources) Cisco Systems Inc., *Cisco LocalDirector Configuration and Command Reference Guide (Software Version 4.2.1)*, Available: http://www.cisco.com/en/US/docs/app_ntwk_services/waas/localdirector/command/v421/reference/LD42_ch03.html

[32] A. Dhananjay and L. Ruan, "PigWin: Meaningful load estimation in. IEEE 802.11 based wireless LANs," in *Proc. IEEE ICC*, Beijing, China, May 2008, pp. 2541–2546.

[33] 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, Jun. 1993.

[34] (Online Sources) Cisco Systems Inc., *How Does Unequal Cost Path Load Balancing (Variance) Work in IGRP and EIGRP?*, Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a008009437d.shtml

[35] M. Lengyel, J. Sztrik, and C. S. Kim, "Simulation of differentiated services in network simulator," Annales Universitatis Scientiarium Budapestinensis de Rolando Eötvös Nominatae, Sectio Computatorica, 2003.

[36] M. Lengyel and J. Sztrik, "Performance comparison of traditional schedulers in DiffServ architectures Using NS," in *Proc. the 16th European Simulation Symposium (ESS)*, Budapest, Hungary, Oct. 2004.

[37] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round robin," IEEE/ACM Trans. Networking, vol. 4, no. 3, pp. 375–385, Jun. 1996.

[38] 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, Jul. 2006.

[39] A. Zinin, Cisco IP Routing, packet forwarding and intra-domain routing protocols. Reading, MA: Addison Wesley, 2002.

[40] C. Hopps, "Analysis of an equal-cost multi-path algorithm," RFC 2992, Nov. 2000.

[41] 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.

[42] 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.

[43] Ja. Kim, B. Ahn, and Ju. Kim, "Multiple path selection algorithm using prime number," in *Proc. the 10th International Conference on Communications Systems (ICCS 2006)*, Singapore, Oct. 2006, pp. 1–5.

[44] J. Kim and B. Ahn, "Next-hop selection algorithm over ECMP," in *Proc. Asia Pacific Conference on Communications (APCC 2006)*, Busan, Korea, Aug. 2006.

[45] Y. Lee and Y. Choi, "An adaptive flow-level load control scheme for multipath forwarding," Lecture Notes in Computer Science, Springer-Verlag, vol. 2093, pp. 771–779, July, 2001.

[46] T. W. Chim, K. L. Yeung, and K.-S. Lui, "Traffic distribution over equal-cost-multi-paths," Computer Networks, vol. 49 (4), Nov. 2005, pp. 465–475.

[47] R. Martin, M. Menth, and M. Hemmkeppler, "Accuracy and dynamics of hash-based load balancing algorithms for multipath Internet routing," in *Proc. IEEE International Conference on Broadband Communications, Networks, and Systems (BROADNETS)*, San José, CA, Oct. 2006.

[48] K. Chebrolu and R. R. Rao, "Bandwidth aggregation for real-time applications in heterogeneous wireless networks," IEEE Trans. Mobile Computing, vol. 5, no. 4, pp. 388–403, Apr. 2006.

[49] J. C. Fernandez, T. Taleb, M. Guizani, and N. Kato, "Bandwidth aggregation-aware dynamic QoS negotiation for real-time video streaming in next-generation wireless networks," IEEE Trans. Multimedia, vol. 11, no. 6, pp. 1082–1093, Oct. 2009.

[50] J. Song, S. Kim, M. Lee, H. Lee, and T. Suda, "Adaptive load distribution over multipath in MPLS networks," in *Proc. IEEE ICC*, Anchorage, Alaska, May 2003, pp. 233–237.

[51] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of Internet traffic engineering," RFC 3272, May. 2000.

[52] 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.

[53] D. Loguinov and H. Radha, "Measurement study of low-bitrate internet video streaming," in *Proc. 1st ACM SIGCOMM Workshop on Internet Measurements*, CA, Nov. 2001, pp. 281–293.

[54] 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 Communications, vol. 30, no.9, pp.1980–1993, Jun. 2007.

[55] N. M. Piratla and A. P. Jayasumana, "Reordering of packets due to multipath forwarding – An analysis," in *Proc. IEEE ICC*, Istanbul, Turkey, Jun. 2006, pp. 829–834.

[56] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," RFC 2330, May. 1998.

[57] C. Demichelis and P. Chimento, "IP packet delay variation metric for IP performance metrics (IPPM)," RFC 3393, Nov. 2002.

[58] (Online Sources) P. Danzig, J. Mogul, V. Paxson, and M. Schwartz. (1995, March). *The Internet Traffic Archive*. Available: http://ita.ee.lbl.gov/index.html.

[59] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, and J. Perser, "Packet reordering metrics," RFC 4737, Nov. 2006.

[60] A. Jayasumana, N. Piratla, T. Banka, and R. Whitner, "Improved packet reordering metrics," RFC 5236, Jun. 2008.

[61] X. Li and L. Cuthbert, "Multipath QoS routing of supporting DiffServ in mobile ad hoc networks," in *Proc. the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN)*, MD, May. 2005.

[62] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated services," RFC 2475, Dec. 1998.

[63] F. Aune, *Cross-Layer Design Tutorial*, Norwegian University of Science and Technology, Trondheim, Norway, Published under Creative Commons License, Nov. 2004.

[64] 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, Jul. 1994.

[65] O. Kremien and J. Kramer, "Methodical analysis of adaptive load sharing algorithms," IEEE Trans. Parallel Distribution Systems. vol. 3, no. 6, pp. 747–760, Nov .1992.

[66] 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.

[67] Z. Tari, J. Broberg, A. Zomaya, and R. Baldoni, "A least flow-time first load sharing approach for distributed server farm," Journal of Parallel and Distributed Computing, vol. 65, no. 7, pp. 832–842, Jul. 2005.

[68] 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," in *Proc. the* 26th *International Communications Satellite Systems Conference* (*ICSSC*), San Diego, CA, Jun. 2008.

[69] 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," in *Proc. IEEE International Conference on Network Infrastructure and Digital Content* (*IC-NIDC 2009*), Beijing, China, Nov. 2009, pp. 236–240.

[70] Sumet Prabhavat, Hiroki Nishiyama, Nirwan Ansari, and Nei Kato, "Effective Delay-Controlled Load Distribution over Multipath Networks," IEEE Trans. Parallel and Distributed Systems. (*Accepted*)

[71] Sumet Prabhavat, Hiroki Nishiyama, Nirwan Ansari, and Nei Kato, "On Load Distribution over Multipath Networks," IEEE Communications Surveys & Tutorials. (*Under Revision and Review*)

[72] G. Almes, S. Kalidindi, and M. Zekauskas, "A One-way Delay Metric for IPPM," RFC 2679, Sep. 1999.

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

[74] (Online Sources) Cisco Systems Inc., *How Does Load Balancing Work?*, Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml

[75] G. Iannaccone, S. Jaiswal, and C. Diot, "Packet reordering inside the sprint backbone," Tech. Report, TR01-ATL-062917, Sprint ATL, Jun. 2001.

[76] F. Bonomi, B. Doshi, J. S. Kaufman, T.P. Lee, and A. Kumar, "A case study of an adaptive load balancing algorithm," Queueing Systems: Theory and Applications, vol. 7, no. 1, pp. 23–49, Nov. 1990.

[77] F. Bonomi and A. Kumar, "Adaptive optimal load balancing in a heterogeneous multiserver system with a central job scheduler," IEEE Trans. Computers, vol. 39, no. 10, pp. 1232–1250, Oct. 1990.

[78] K. P. Bubendorfer, “Resource Based Policies for Load Distribution,” Master's Thesis of Victoria University of Wellington, Aug. 1996.

[79] E. S. H. Hou, N. Ansari, and H. Ren, “A Genetic Algorithm for Multiprocessor Scheduling,” IEEE Trans. Parallel and Distributed Systems, vol. 5, no. 2, pp. 113–120, Feb. 1994.

[80] B. Fortz and M. Thorup, “Internet Traffic Engineering by Optimizing OSPF weights,” in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, pp. 519–528.

[81] Y. Wang, Z. Wang, and L. Zhang, “Internet Traffic engineering without full mesh overlaying,” in *Proc. IEEE INFOCOM*, Anchorage, Alaska, Apr. 2001, pp. 565–571.

[82] H. Abrahamsson, B. Ahlgren, J. Alonso, A. Andersson, and P. Kreuger, “A multi-path routing algorithm for IP networks based on flow optimization, ” in *Proc. International Workshop on Quality of Future Internet Services (QofIS'02)*, Zurich, Switzerland, Oct. 2002.

[83] W. Fischer and K. Meier-Hellstern, “The Markov-Modulated Poisson Process (MMPP) cookbook,” Performance Evaluation, vol. 18 (2), pp. 149–171, Sep. 1993.

[84] A. Sridharan, R. Guerin and C. Diot, “Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF/IS-IS Networks,” IEEE/ACM Trans. Networking, vol. 13, no. 2, pp. 234–247, Apr. 2005.

[85] R. Chandra, P. Bahl, and P. Bahl, "MultiNet: Connecting to multiple IEEE 802.11 networks using a single wireless card," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 882–893.

[86] 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.

# List of Publications

**Journal Papers**

- Sumet Prabhavat, Hiroki Nishiyama, Nirwan Ansari, and Nei Kato, "Effective Delay-Controlled Load Distribution over Multipath Networks," In IEEE Transactions on Parallel and Distributed Systems. (*Accepted*)

- Sumet Prabhavat, Hiroki Nishiyama, Nirwan Ansari, and Nei Kato, "On Load Distribution over Multipath Networks," In IEEE Communications Surveys & Tutorials. (*Under Revision and Review*)

**International Conference Papers**

- 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," In Proceeding of the 26th International Communications Satellite Systems Conference (ICSSC), San Diego, CA, Jun. 2008.

- Sumet Prabhavat, Hiroki Nishiyama, and Nei Kato, "Delay-Minimized Load Distribution for Multipath Networks," In Proceeding of the 1st Student Organizing International Mini-Conference on Information Electronics Systems (SOIM-GCOE08), Sendai, Japan, Oct. 2008. (*Poster Session*)

- 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," In Proceeding of the IEEE International Conference on Network Infrastructure and Digital Content (IEEE IC-NIDC), Beijing, China, Nov. 2009. (*Received the Best Paper Award*)