On Load Distribution over Multipath Networks

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

***Abstract*—Efficient utilization of network resources, provided by multiple interfaces available on today devices, is critical in facilitating parallel connections through multiple paths. Load distribution strategies in using multiple interfaces for simultaneous data transmission have been studied. This paper presents a thorough literature review of various existing load distribution models, and classifies them in terms of their key functionalities such as traffic splitting and path selection. 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 analyze various examples of existing models, and then compare and identify their exhibited advantages as well as shortcomings.**

***Index Terms*—Load Distribution, Load Balancing, Multipath Forwarding, Traffic Engineering, Traffic Splitting**

I. Introduction

THE DEMAND for a wide variety of network services has been the major driving force for innovation and development of various networking technologies. Network capacity provisioning and Quality of Service (QoS) guarantees are key issues in meeting this demand. The presence of several physical/logical interfaces incorporated with a multipath routing/forwarding protocol allows users to use multiple paths in establishing simultaneous connections. The exploitation of multiple paths no longer aims only at circumventing single point of failure scenarios but also focuses on facilitating network provision, where its effectiveness is indeed essential to maximize high quality network services and guarantee QoS at high data rates [1], [2]. Bandwidth aggregation and network-load balancing are two major issues that have attracted tremendous amount of research, and a number of load distribution approaches have been proposed.

Before plunging into details of load distribution models, for the sake of completeness, we discuss multipath configurations that can be established in several different ways, as shown in Fig. 1. Fig. 1(a) and 1(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(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 [3]. A special routing technique is required at a source or a gateway to establish multiple path routing. In the Internet, one of the most well-known routing techniques is Equal-Cost Multi-Path (ECMP) routing [4], [5] which is currently supported by Internet routing protocols such as Open Shortest Path First (OSPF) [6], Routing Information Protocol (RIP) [7], [8], and Enhanced Interior Gateway Routing Protocol (EIGRP) [9]. ECMP routes packets along multiple paths of equal cost. In Multi-Protocol Label Switching (MPLS) networks [10], 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) [11] or Resource Reservation Protocol-Traffic Engineering (RSVP-TE) [12]. Load balancing structures without dynamic traffic engineering can incur heavy use of multipath routing [13], [14]. Various kinds of dynamic traffic engineering techniques for load balancing over multiple paths, such as [15], [16], [17], and some others overviewed in [18], [19], have been proposed. Fig. 1(c) is a special case of Fig. 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) [20], and Wireless Fidelity (IEEE 802.11). On the other hand, inverse multiplexing [21] depicted in Fig. 1(d) can be considered as an abstraction of Fig. 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 [22]. Wide Area Multi-Link PPP (WAMP) [23], strIPe [24], and Dynamic Hashing with Flow Volume (DHFV) [25] are implementations of inverse multiplexing. Fig. 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) [26] and Multi-path Source Routing (MSR) [27] developed based on Dynamic Source Routing (DSR) [28], and Ad hoc On-demand Distance Vector - Multipath (AODVM) [29] and Ad hoc On-Demand Multipath Distance Vector (AOMDV) [30] developed from Ad hoc On-Demand Distance Vector (AODV) [31] are notable multipath routing protocols for MANETs. In [32], QoS is taken into consideration in routing. For satellite mesh network consisting of non-geostationary satellites, Explicit Load Balancing (ELB) [33] has been developed to distribute traffic among multiple different links in order to avoid traffic convergence.

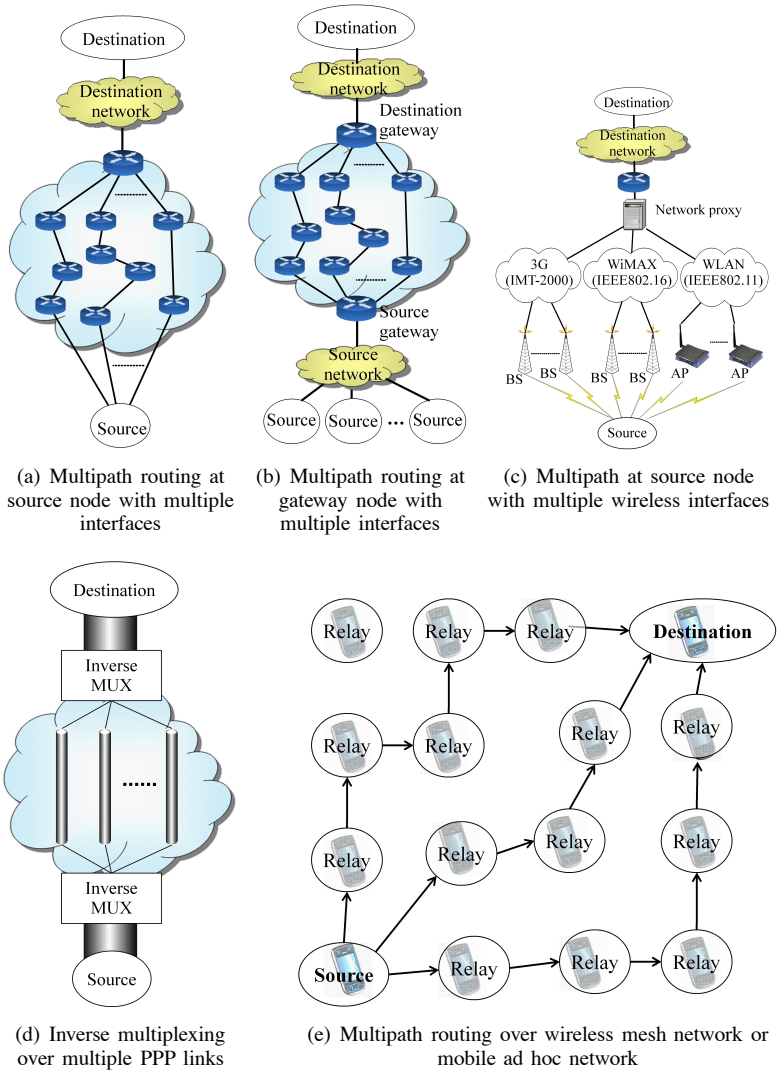
Manuscript received 28 January 2010; revised 31 October 2010, 6 March 2011, and 29 June 2011.

S. Prabhavat is with the Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand e-mail: sumet@it.kmitl.ac.th.

H. Nishiyama and N. Kato are with the Graduate School of Information Sciences, Tohoku University, Sendai, JAPAN e-mail: bigtree@it.ecei.tohoku.ac.jp and kato@it.ecei.tohoku.ac.jp.

N. Ansari is with the Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA e-mail: Nirwan.Ansari@njit.edu.

Digital Object Identifier 10.1109/SURV.2011.082511.00013



(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

Fig. 1. Examples of various multipath configurations

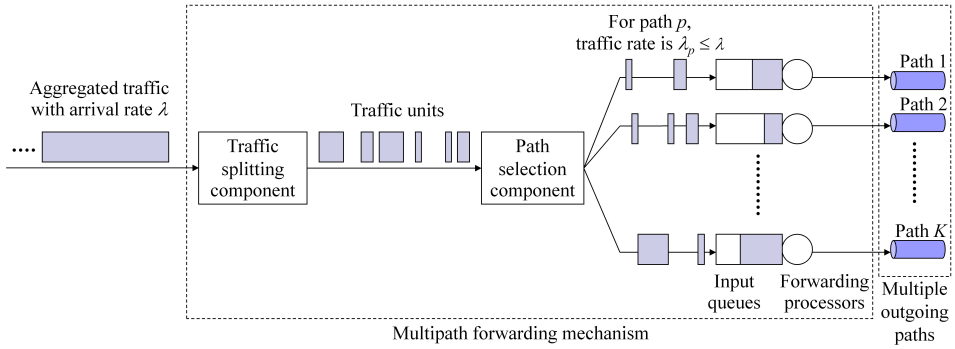
As mentioned above, there exist different networks with various environments in establishing multiple paths, and multiple paths can be established by using a totally different technology in each situation. Since we focus on load balancing over multiple paths, we do not address routing to establish multiple paths. In other words, multiple paths for load balancing are assumed to be already established by routing techniques. In this paper, the generalized multipath forwarding mechanism is first described in Section II while various existing load distribution models along with key functional components in multipath forwarding are introduced in Section III. Significant performance issues and criteria in load distribution such as ability to prevent load imbalance, bandwidth utilization efficiency in each path, and ability to maintain packet ordering, are presented in Section IV. The performance of existing models is compared and evaluated based on qualitative analysis and simulation results in Section V and Section VI, respectively. Finally, we conclude this paper in Section VII.

II. Multipath Forwarding Mechanisms

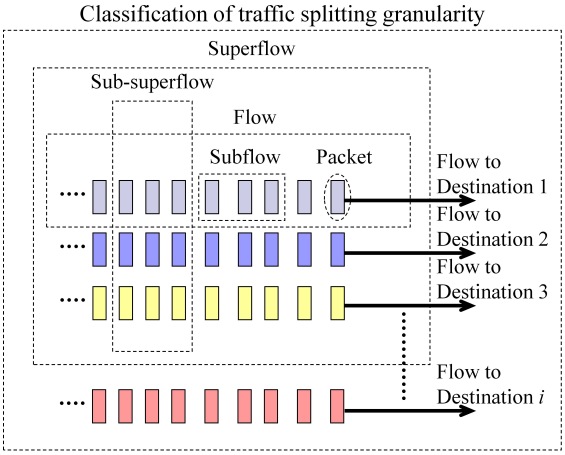
The important role of load distribution is engineered by the traffic splitting and path selection, which are the key components of multipath forwarding and the focus of this paper. Note that separately analyzing these two components of a multipath forwarding mechanism is one of our main contributions that are expected to help readers understand load distribution models. After having described the general multipath forwarding mechanism, different types of traffic units and different path selection schemes will be discussed.

A. Basic Multipath Forward

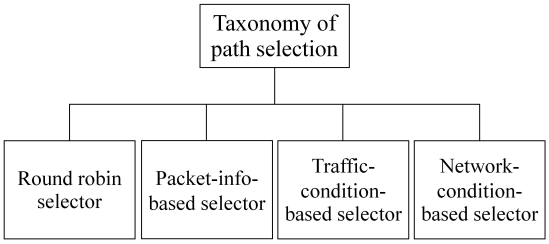
Fig. 2(a) illustrates the functional components of multipath forwarding: traffic splitting and path selection. The traffic splitting component splits the traffic into traffic units, each of which independently takes a path, which is determined by the path selection component. If the forwarding processor 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.



(a) Functional components of the multipath forwarding mechanism



(b) Classification of traffic splitting granularity



(c) Taxonomy of path selection

Fig. 2. Multipath forwarding mechanism and its internal functional components

B. Traffic Splitting

By the traffic splitting component, aggregated traffic from traffic sources is split into several traffic units [34], where the constitution of a traffic unit depends on the level of splitting granularity. The traffic splitting classification is illustrated in Fig. 2(b).

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 determined from destination addresses stored in packet headers, are taken into consideration in splitting. All packets heading for the same destinations are grouped together; each group is defined as a unit of flow with a unique flow identifier. Splitting traffic at this level can maintain packet ordering since path selection for all packets in the same flow is identical. The path selection for each flow is made independently. 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, packet header information such as source address, type of service, and protocol number can be used [35].

In subflow-level traffic splitting, a flow of packets heading for the same destination is allowed to be split into subflows (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 [36] 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 for load balancing in the Internet. 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 [37].

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 a subflow and a 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

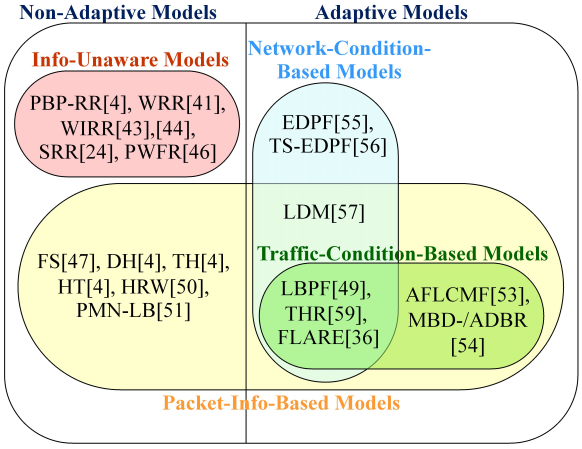


Fig. 3. Load distribution model classification

in a sub-superflow) can be taken into account in the traffic splitting.

C. Path Selection

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. In a load distribution model with packet-level traffic splitting, the selection is made independently for each packet, while, in that with the other traffic splitting (i.e., flow, subflow, superflow, and sub-superflow-level traffic splitting), the selection will be made similarly for all packets of the same traffic unit. Most path selection schemes can be categorized into four types as shown in Fig. 2(c) 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 [38], [39] is rather simple with the computational complexity of $O(1)$, requiring no additional network information for path selection.

In packet-info-based selector (PacketInfo), 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 [40], and are selected depending upon control objectives.

In network-condition-based selector (NetCon), network conditions such as path delay, path loss, and queue length are used to determine the output path, according to the goal of load balancing.

III. Existing Models

Existing load distribution models can be classified into two categories, namely, non-adaptive and adaptive models which are further classified in the first and the second subsection, as illustrated in Fig. 3. Various examples of load distribution models are investigated in terms of their functionalities, characteristics as well as internal functional components. Then, they are summarized in the last subsection.

A. Classification of 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.

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. Their major advantages and drawbacks are summarized in Table I.

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

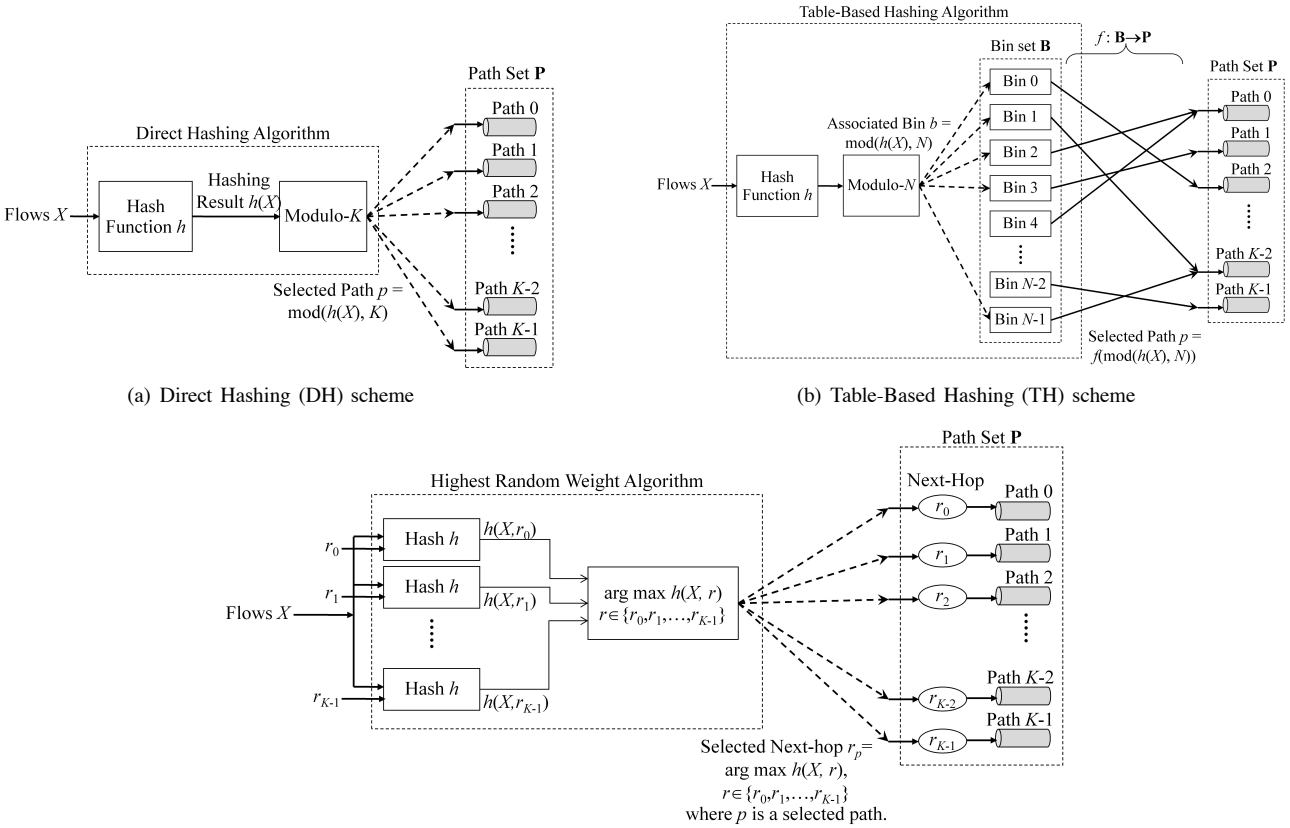
PBP-RR has been implemented in several applications, e.g., ECMP routing and inverse multiplexing. The first example is incorporated in packet-switched networks while the latter is in multiple point-to-point networks. Since PBP-RR implements the packet-based round-robin path scheduling [4], it achieves simplicity and starvation-free (i.e., no idle path exists while a packet is waiting to be sent) and causes 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 [41] is implemented to support heterogeneous multiple paths [9], [42]. 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 [9] and MSR [27]. In WRR, load imbalance can occur owing to variation in the size of packets. Also, it can occur because of 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 [43], [44] possesses characteristics almost similar to those of 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



(a) Direct Hashing (DH) scheme

(b) Table-Based Hashing (TH) scheme

(c) Highest Random Weight (HRW) scheme

Fig. 4. Functional components of the well-known hash-based algorithms for Internet load balancing

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 [24] is based on a modified version of Deficit Round Robin (DRR) [45], which is a modified WRR. 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. Each path is associated with a deficit counter and a quantum of service, measured in bytes, proportional to the bandwidth of the path. The deficit counter representing the difference between the desired and actual loads allocated to each path is taken into account in the path selection. At the beginning of each round, the deficit counter is increased by the given 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 has begun. SRR has been implemented for load balancing in packet-switched networks, as a part of strIPe protocol [24].

Packet-By-Packet Weighted Fair Routing (PWFR)

PWFR [46] 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 packets 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 $O(K)$; 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) Packet-Info-Based (Non-Adaptive) Models: Packet reordering is the major problem of the info-unaware models. Selecting the same path for all packets having the same destination address can solve the problem. To do so, packet information is required for path selection. This idea has been incorporated in [47] and has also been studied in hash-based schemes [5], [37], [48], as summarized in Table I and detailed as follows.

TABLE I
SUMMARY OF NON-ADAPTIVE LOAD DISTRIBUTION MODELS

Model	Advantages and enhancement	Remaining problems and limitations
Info-unaware Models		
PBP-RR [4]	Simple. No communication overhead.	Not applicable for multiple paths with different characteristics. No mechanism to prevent packet reordering.
WRR [41]	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 [43], [44]	Prevent the continuous use of a particular path.	Similar to WRR.
SRR [24]	Similar to WRR, but byte-based deficit counter allows to cope with variation in packet size distribution.	No mechanism to prevent packet reordering.
PWFR [46]	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 [47]	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 [4]	Simple. No communication overhead.	Load imbalance caused by variation in flow size distribution and non-uniformity of hash distribution. High disruption.
TH [4]	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 [4]	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 [50]	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 [51]	Low disruption and low complexity.	Load imbalance caused by variation in flow size distribution and non-uniformity of hash distribution.

Fast Switching (FS)

FS [47] 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, all packets are sent via the same path. 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. FS can balance the number of flows distributed among the paths. However, FS cannot deal with skewness of flow size distribution, which can cause load imbalance. 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 entry is removed. As a consequence, the path for the oldest flow may change resulting in packet reordering. 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. Since a path for forwarding the packet is determined by looking up in a flow-path mapping table, it has computational complexity of $O(K)$. This can create scalability issues when the number of flows or paths increases. In FS, when a path is removed, 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 [5],[48], (which will be further elaborated in the next section) 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 [4], [5], [48]. It performs hash-based load balancing for ECMP routes. Its functional components are illustrated in Fig. 4(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, 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 [37], [49]. 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 [4] is a hash-based load balancing scheme in ECMP

routing. Its functional components are illustrated in Fig. 4(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 [37]; 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 [4], which is a load balancing scheme incorporated in ECMP routing, possesses characteristics almost similar to those of TH in Fig. 4(b) except the mapping table (f). It partitions the hash result space into regions. Each region is a set of flow IDs which will be routed via a path. Customizing the size (i.e., s_p) of a region corresponding to each path (i.e., path p) controls the number of flows sent to the path. The ratio among region sizes (i.e., s_1: s_2: ...: s_K) is the load ratio among K paths. Probability of each path selected is determined by the region size [5], [48]. For example, in order to achieve equal load sharing, the hash result space is equally partitioned; all K regions have the same size. 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 [50] 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 path identifier (r_i) which is the next hop address of path i, as illustrated in Fig. 4(c). Among all paths whose next hop addresses are r_0 to r_{K-1}, the path p with the highest random weight is selected. 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 [5], [48], but it has a higher computational complexity, $O(K)$. Lookup performance will degrade when the number of flows grows large.

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

PMN-LB [51], [52] 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.

B. Classification of 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.

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. Their advantages and drawbacks are summarized in Table II.

Adaptive Flow-Level Load Control Scheme for Multipath Forwarding (AFLCMF)

AFLCMF [53] is a flow-aware adaptive multipath load control scheme for load balancing in packet-switched networks. AFLCMF allows us to distribute traffic among multiple paths in a pre-defined ratio which is a desired load ratio. The desired load ratio and a measured packet arrival rate are taken into account in determining a rate threshold used for flow classification. 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 the rate threshold will be sent via path 1; otherwise, it will be sent via path 2. When a packet arrival rate changes, the rate threshold is adjusted. Varying the rate threshold affects the number of flows sent via each path, and thus controls the ratio of load among the multiple paths. Therefore, load assigned on each path can be 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 $O(K)$, increase when the numbers of active flows and parallel paths increase, respectively.

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

TABLE II
SUMMARY OF ADAPTIVE LOAD DISTRIBUTION MODELS

Model	Advantages and enhancement	Remaining problems and limitations
Traffic-condition-based Adaptive Models		
AFLCMF [53]	Load sharing ratio can be controlled by a predetermined parameter.	Since adaptation is invoked for all packet arrivals, it is possible to cause flow redistribution and packet reordering.
MBD-/ADBR [54]	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 [55]	Selecting a path which can deliver a packet to the destination at the earliest time can reduce packet delay.	Selecting a path having the smallest delay can cause a risk of packet reordering.
TS-EDPF [56]	Scheduling packets on each path based on time slot related to bandwidth negotiated from a QoS server can reduce packet delay and guarantee quality of service.	Similar to EDPF.
LDM [57]	A path with lower utilization and smaller hop-count has a higher 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 [49]	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 [59]	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 [36]	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.

MBD-/ADBR [54] is a variant version of TH. In contrast, the flow-to-path mapping table f illustrated in Fig. 4(b) can be dynamically changed. The number of packets in each superflow (associated with a corresponding bin) 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 $O(K)$. In each control phase, processing time increases as the numbers of superflows and paths increase.

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. Table II presents their major advantages and drawbacks.

Earliest Delivery Path First (EDPF)

EDPF [55] 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 $O(K)$.

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

TS-EDPF [56], 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 [57] is a load distribution model relying on the traffic engineering concept [58], designed for MPLS networks [10]. 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 $O(K)$.

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 [49], [59], and packet order preservation [36]. Their advantages and drawbacks are summarized in Table II.

Load Balancing for Parallel Forwarding (LBPF)

W. Shi, *et al*. [49] 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 [49], 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 flows 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 $O(K)$, increase as the numbers of active flows and parallel paths increase, respectively.

Table-Based Hashing with Reassignment (THR)

THR [59] is similar to TH but the flow-to-path mapping table f illustrated in Fig. 4(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, β, which determines the priority between improving load imbalance and preventing packet reordering. With $\beta \rightarrow 0$, THR aims to reduce the load imbalance by moving the largest superflow. On the other hand, with $\beta \rightarrow \infty$, 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 β, 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 $O(K)$.

Flowlet Aware Routing Engine (FLARE)

FLARE [36] 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 $O(K)$.

The conditional splitting of flows is a key property of FLARE. For a smaller threshold, the deviation from the desired load distribution can be decreased at the price of higher risk of packet reordering, and vice versa. In order to guarantee that the two consecutive packets can be assigned to different paths without the risk of packet reordering, the threshold has to be larger than the value of the mean time before switch-ability (MTBS), which is the maximum delay difference among all available parallel paths. To estimate the

value of MTBS, FLARE periodically executes an estimation technique, e.g., ping operation, to measure the round trip delay of each path and calculates the maximum delay difference among the parallel paths from the measured delays; it uses the obtained value as an estimate of MTBS. The performance of FLARE largely depends on the estimation accuracy. An overestimation error causes a flow not to be split even if it should be, thus reducing the opportunity of splitting. On the other hand, an underestimation error causes a flow to be split more than necessarily, thus causing packet reordering. Moreover, in the bursty traffic environment, a sudden increase in the packet arrival rate can cause underestimation errors. While more frequent measurements may be able to reduce the estimation error, they incur communication overhead, thus consuming additional bandwidth resources.

C. Summary

Existing load distribution models are 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. 3. Table I summarizes advantages and limitations of non-adaptive load distribution models. Info-unaware models make a raw decision on distributing traffic without taking external information into account. A common major drawback of models in this class is their inability to maintain packet ordering. Non-adaptive packet-info-based models making a decision on path selection based on packet information select the same path for all packets having the same destination address in order to solve the packet reordering problem. Adaptive models require traffic condition estimated from the incoming traffic and network condition measured by network measurements. Table II summarizes their advantages and limitations. For highly skewed flow size distribution, traffic load cannot be balanced by non-adaptive models. Adaptive path selection based on traffic condition can mitigate this problem. 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. Moreover, conditionally splitting only a traffic flow having its packet inter-arrival time larger than some threshold can mitigate the packet reordering problem.

IV. Performance Issues

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

A. Load Imbalance

Appropriate load sharing can be achieved when the load is assigned on each path properly according to the desired load derived from the capacity of the path in terms of, e.g., bandwidth capacity and buffer size. The difference between the desired and actual loads on a particular path is generally referred to as load balancing deviation, and also called deficit load in queuing analysis. 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.

According to the analytical results reported in [36], the upper bound of the probability that, at time t, a path p has a deficit load, $D_p(t)$, larger than a certain threshold, ξ, in absolute values can be expressed as follows:

$$Pr[|D_p(t)| > \xi] < \frac{1}{4\xi^2 E[N(t)]}(\gamma^2 + 1) \quad (1)$$

E[$N(t)$] is the expected number of traffic units induced during the interval (0, t], and γ is the Coefficient of Variation (CV) of the size of traffic-unit. Equation (1) clearly represents the fact that a smaller traffic unit contributes better load balancing because it tends to lead to a smaller γ and a larger value of $N(t)$. This result matches to the proof given in [50], [53] that the variance of sizes of one traffic unit must be finite to minimize the load balancing deviations over all paths. We can surely understand the reason why load distribution models with packet-level traffic splitting can achieve near perfect load balance in minimizing load balancing deviation. While the variation of the packet size distribution is bounded by network parameters such as the maximum packet size, that of flow size has no such bound.

B. Inefficient Bandwidth Utilization

If traffic load is perfectly balanced such that all outgoing paths are busy or idle at the same time, the load distribution system is work-conserving where bandwidth utilization is maximized (i.e., no bandwidth loss). Otherwise, it is a non-work-conserving system; at least one path has no load while the other paths are busy, thus resulting in bandwidth loss on idle paths. The non-work-conserving idle time is used as a metric in performance evaluation. Non-work-conservation is affected by the variation in the size of the traffic units and the path determination policy. A large variation in the size of the traffic units or a path selection unaware of the path working ratio results in 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.

C. Degree of Flow Redistribution

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 two successive 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. In this paper, 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. The flow redistribution can lead to the important problem such as packet reordering which will be described next.

D. Packet Reordering

In the Internet, packet reordering is not a sporadic event [60]. Actually, the packet reordering problem significantly impairs TCP traffic flows (which are mostly found in the Internet) [60], real-time traffic flows, and multimedia traffic flows [61]. 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 [62], [63]. 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 [64], [65]. 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 [66]. These trade-offs need to be taken into account in mitigating the packet reordering issue.

E. Communication Overhead

To estimate the network condition, some 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 in the network. The additional traffic not only decreases 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.

F. Computational Complexity

Computational complexity is defined as the computational load required to determine the outgoing path for each arrived packet. A simple path selection algorithm using constant-sized table, independent of the values of parameters such as the number of available paths, has the computational complexity of $O(1)$, whereas the algorithm of finding a path from a list of K paths has the computational complexity of $O(K)$. For example, the computational load incurred by HRW is higher than that caused by DH and TH.

G. Implementation Complexity

Implementation complexity must be considered for realization of any technology. Likewise, this is an important issue for load balancing. For example, measurement of path delays in FLARE, measurement of traffic rate in LBPF, AFLCMF, and MBD-/ADBR, and packet counter as well as packet inter-arrival timer in THR are critical components for performance improvement, but they incur significant computational complexities and overheads. To be realizable in real networks, installations of extra components and modifications of existing ones should be minimized. Considering the implementation of FLARE into IP networks at a source as an example, there are two different implementation methods for path-delay measurement function, i.e., the use of the round trip time measurement mechanism already equipped in the upper layer protocol such as TCP, and the utilization of Internet control message protocol (ICMP) echo request/reply mechanism in the IP layer. In this example, the difference between the former method which requires cross layer implementation [67] and the latter one which needs inter-protocol implementation should be considered and evaluated.

V. Qualitative Comparisons

The comparative performance of existing load distribution models is summarized in Table III. 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. In the following, except the absolute performance evaluations in adaptability, communication overhead, computational complexity, and implementation complexity, the relative performance comparisons in the load balancing efficiency, bandwidth utilization efficiency, packet order preservation, degree of flow redistribution, and degree of disruption are further discussed.

A. Load Balancing Efficiency

Table III shows comparisons in load balancing efficiency achieved by packet-based load distribution models and the other load distribution models. Since packet-based load distribution models have the smallest traffic unit, with any path selection, they are likely to achieve load balancing. However,

TABLE III
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 redistribution	Degree of disruption	Communica-tion overhead	Computa-tional complexity
Info-unaware Models										
PBP-RR[4]	Packet	RR	n/a	★	★★★	n/a	High	n/a	No	$O(1)$
WRR[41]	Packet	RR,TraffCon (packet counter)	n/a	★★★	★★☆	n/a	High	n/a	No	$O(1)$
WIRR[43],[44]	Packet	RR,TraffCon (packet counter)	n/a	★★★	★★★	n/a	High	n/a	No	$O(1)$
SRR[24]	Packet	RR,TraffCon (deficit byte counter)	n/a	★★★	★★☆	n/a	High	n/a	No	$O(1)$
PWFR[46]	Packet	TraffCon (deficit byte counter)	n/a	★★★	★★☆	n/a	High	n/a	No	$O(K)$
Packet-info-based (non-adaptive) Models										
FS[47]	Flow	PacketInfo, RR (for a new flow)	n/a	★	★	★★★	No	High	No	$O(K)$
DH[4]	Flow	PacketInfo	n/a	★	★	★★★	No	High	No	$O(1)$
TH[4]	Superflow	PacketInfo	n/a	★	★	★★★	No	High	No	$O(1)$
HT[4]	Superflow	PacketInfo	n/a	★	★	★★★	No	Medium	No	$O(1)$
HRW[50]	Flow	PacketInfo	n/a	★	★	★★★	No	Low	No	$O(K)$
PMN-LB[51]	Flow	PacketInfo	n/a	★	★	★★★	No	Low	No	$O(1)$
Traffic-condition-based Adaptive Models										
AFLCMF[53]	Subflow	PacketInfo, TrafficCon (when traffic condition changes)	Yes	★★☆	★★	★★	Medium	n/a	No	$O(K)$
MBD-/ADBR [54]	Sub-superflow	PacketInfo, TrafficCon (when splitting condition is satisfied)	Yes	★★☆	★★☆	★★	Medium	n/a	No	$O(K)$
Network-condition-based Adaptive Models										
EDPF[55]	Packet	NetCon	Yes	★★★	★★★	★☆	High	n/a	Yes	$O(K)$
TS-EDPF[56]	Packet	NetCon	Yes	★★★	★★★	★☆	High	n/a	Yes	$O(K)$
LDM[57]	Flow	PacketInfo (for existing flow), NetCon (for a new flow)	Yes	★☆	★☆	★★★	No	Low	Yes	$O(K)$
Traffic and Network-conditions-based Adaptive Models										
LBPF[49]	Subflow	PacketInfo, TrafficCon (when load adaptation algorithm is activated)	Yes	★☆☆	★☆☆	★☆☆	Low-Medium	n/a	No	$O(K)$
				Trade-off *						
THR[59]	Sub-superflow	PacketInfo, TrafficCon–NetCon (when splitting condition is satisfied)	Yes	★★☆	★★☆	★★☆	Medium-High	n/a	No	$O(K)$
				Trade-off *						
FLARE[36]	Subflow	PacketInfo, TrafficCon (when delay-based splitting condition is satisfied)	Yes	★★☆	★★☆	★★★	Medium-High	n/a	Yes	$O(K)$
				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.

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 is selected.

In flow-based models, load imbalance can be attributed to large variation of flow size distribution. The flow-based models which can follow dynamic changes in traffic/network conditions can mitigate the load imbalance problem, by splitting a flow into subflows, in order to reduce variation in the size of the traffic units, and by switching a path in order to distribute traffic load. The small traffic unit and intelligent path selector are preferred for optimizing load balancing. LDM balances load by using an adaptive path selector. It 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 caused by 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. It can mitigate the load imbalance problem caused by 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 and under-utilized paths by moving some flows among the 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-utilized. Therefore, MBD-/ADBR is likely to achieve better load balancing as compared to THR. 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. It can achieve perfect load balance efficiency if its parameters are chosen such that a flow is split into single packets. However, when the 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 increases the load balancing deviation.

B. Bandwidth Utilization Efficiency

Splitting traffic into single packets causes minimal non-work-conserving idle time. Table III shows comparisons in bandwidth utilization efficiency. Packet-based models can achieve a small bandwidth 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.

Usually large variation in flow size distribution affects the performance of flow-based models. While a particular path is being used to forward a very large flow, other paths (having already finished forwarding shorter flows) are idle, thus resulting in bandwidth loss. In addition, lack of adaptability to current network condition exacerbates this problem 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, when the network utilization is high, since LDM does not allow a flow to change path, 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 thus be reduced. THR and MBD-/ADBR always select the most under-utilized path; bandwidth loss can thus be reduced.

C. Degree of Flow Redistribution

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., smallest/largest 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. Table III show the comparisons mentioned above.

D. Packet Order Preservation

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. 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 reordering, 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. 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. 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 re-ordering. 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. These comparisons are presented in Table III.

VI. Simulation-based Performance Evaluations

We conducted extensive network simulations by using the raw traffic traces obtained from the real networks to evaluate the performance of the multipath forwarding mechanisms in various load distribution models. The comparisons are presented following the explanation of our simulation setup.

A. Simulation Method

Various traffic traces available online [68], characteristics of which are listed in Tables IV and V, are used for the simulations. For each round of a simulation, input traffic is generated according to each of 8 datasets of 1-hour long packet traces. Each traffic trace has a different mean flow size and coefficient of variation (CV) in flow size distribution (i.e., the probability distribution of the number of packets in each flow for all flows), as illustrated in the tables. Therefore, traffic with various characteristic is generated. For example, traffic generated from trace D1 having the largest CV has the largest variation in flow size distribution; the number of packets per flow varies greatly from flow to flow. The input traffic is forwarded via a multipath network, where the number of parallel paths is 3 (K=3) and the input queue of each path has infinite buffer size. For the generated input traffic

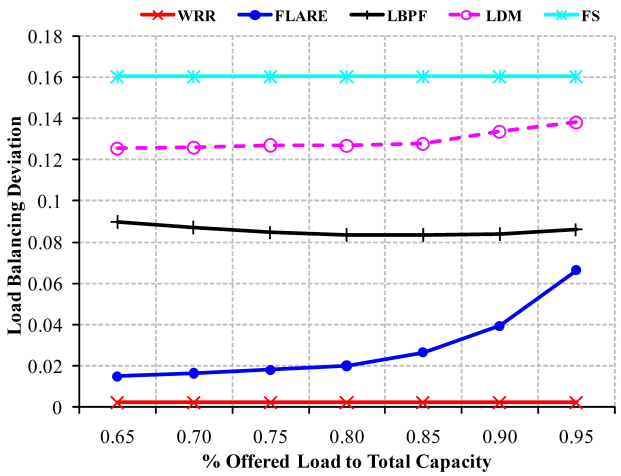


Fig. 5. Comparison in load balancing efficiency

which has a certain mean (packet) arrival rate, the service time for each packet is assumed to be exponentially distributed where the mean service rates of all forwarding processors are identically chosen such that the ratio of the mean offered load (i.e., mean arrival rate) to mean service rate are 0.65, 0.70, . . . , 0.90, and 0.95, respectively. In the conducted simulations, we choose exemplary models from each class having distinctively different functionalities. WRR and FS are examples of non-adaptive load distribution models, whereas LDM, LBPF, and FLARE can adjust their functionalities based on the obtained information such as traffic and network conditions. In the evaluation of the exemplary models, we collect measured results (from the 56 simulation scenarios per model) and compute statistical results.

B. Load Balancing Efficiency

In the evaluation, we calculate load balancing deviation in each second from the measured results. Fig. 5 illustrates the comparisons in load balancing efficiency (averaged among all traces) of the exemplary 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. However, when network utilization increases, the number of packets to be shifted increases while a path to accommodate the packets tends to have less amount of deficit load; load balancing deviation increases in LDM. 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's splitting only aggressive flows, FLARE can better reduce load balancing deviation. However, in FLARE, when network utilization increases to the high load condition (higher than 85

C. Bandwidth Utilization Efficiency

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

TABLE IV
PACKET-LEVEL PROFILE OF TRAFFIC TRACES [68]

Trace ID	Trace name	# Packets	Packet arrival rate (packets/second)		
			Mean	Min.	Max.
D1	dec-pkt-1.tcp	2153462	598.05	140	1917
D2	dec-pkt-1.udp	829759	230.46	103	448
D3	dec-pkt-2.tcp	2661931	739.32	259	1706
D4	dec-pkt-2.udp	805802	223.81	89	468
D5	dec-pkt-3.tcp	2873589	798.07	58	1530
D6	dec-pkt-3.udp	1035457	287.59	18	520
D7	dec-pkt-4.tcp	3862336	1072.71	232	1931
D8	dec-pkt-4.udp	1187454	329.81	69	460

TABLE V
FLOW-LEVEL PROFILE OF TRAFFIC TRACES [68]

Trace ID	Over all sumulation time		In each second						
	# Different flows	Mean flow size	Flow rate (flows/second)			# Flows having size larger than		Mean flow size	Coefficient of variation (CV) in flow size distribution
			Mean	Min.	Max.	100 Packets	1000 Packets		
D1	7559	284.89	117.66	49	181	1717	324	5.46	0.0795
D2	38032	21.82	145.23	77	209	996	109	1.60	0.0250
D3	5865	453.87	137.89	77	204	1397	365	5.58	0.0385
D4	31491	25.59	135.16	68	206	949	101	1.69	0.0397
D5	12903	222.71	175.32	44	247	2842	533	4.66	0.0288
D6	62713	16.51	161.99	16	265	1205	71	1.81	0.0302
D7	12710	303.88	184.50	90	269	2651	621	5.96	0.0345
D8	58025	20.46	174.85	50	257	1175	83	1.90	0.0320

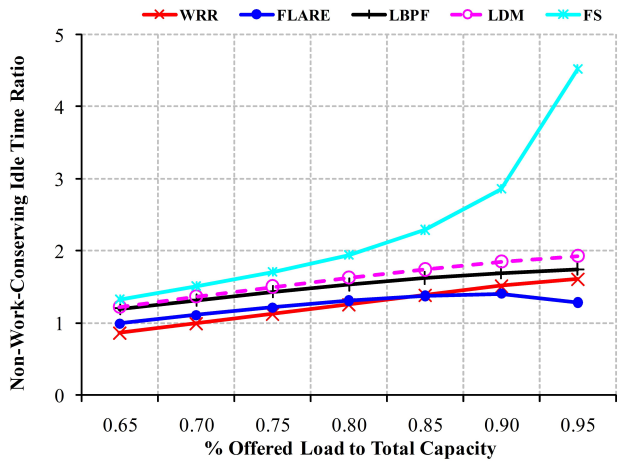


Fig. 6. Comparison in efficiency of bandwidth utilization

busy) to define the metric to evaluate bandwidth utilization efficiency. 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. This is to compare bandwidth loss incurred under a multipath network to that incurred under a single path network. In the best condition, this ratio should be less than 1, implying that bandwidth loss in a multipath network is lower than that in a single path network. The higher ratio indicates worse bandwidth utilization efficiency because of more bandwidth loss.

As described in the previous section that 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 it implies bandwidth loss on idle paths. Fig. 6 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. In contrast, LDM with adaptability to network conditions

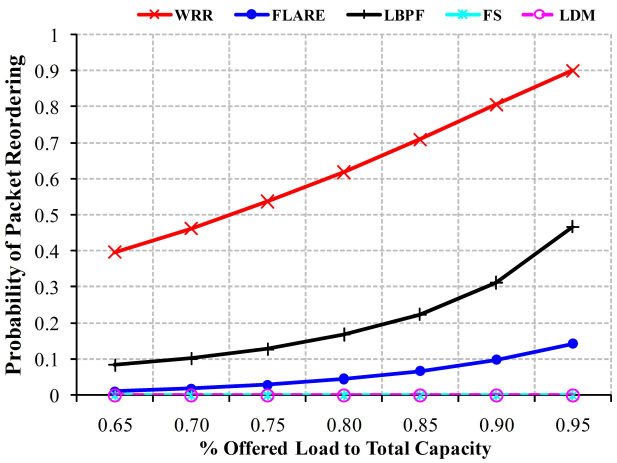


Fig. 7. Comparison in packet order preservation

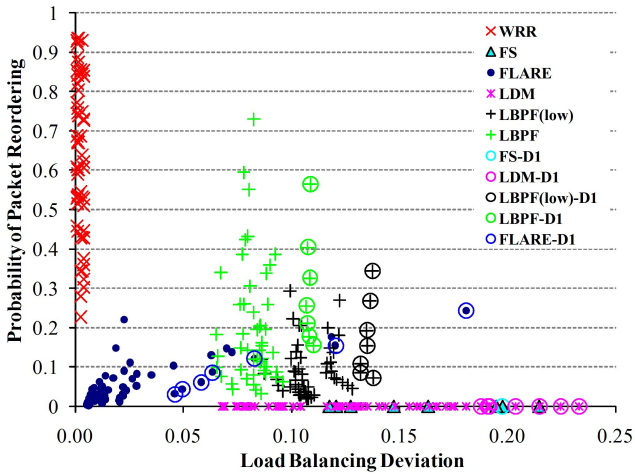


Fig. 8. Load balancing deviation vs. packet reordering

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.

D. Packet Order Preservation

The probability of packet reordering is derived from the probability that the reordering buffer is occupied by arrived packets which have to wait for late packets [69], [70], i.e., ratio between the accumulated number of packets stored in the buffer and the total number of packets. Switching the path of a flow can cause reordering of packets belonging to the flow if the newly selected path has a different delay. As demonstrated in Fig. 7, WRR incurs a high risk of packet reordering. The risk of packet reordering is much higher when network utilization increases. In FS and LDM, there is no risk of packet reordering. With an adaptive load distribution algorithm, LBPF and FLARE lose ability to completely prevent packet reordering. In LBPF, when network utilization increases, the splitting rate increases, thus causing a significant increase of the risk of packet reordering. In FLARE, splitting only flows, which are not expected to incur packet reordering, can maintain a low risk of packet reordering.

E. Packet Order Preservation vs. Load Balancing and Bandwidth Utilization Efficiencies

Figs. 8 and 9 illustrate performance trade-offs between load balancing and bandwidth utilization efficiencies, on the one hand, and prevention of packet reordering, on the other hand. FS and WRR are two extreme cases where each represents the opposite case. Points of FS lie on the x-axis while those of WRR are close to the y-axis. 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

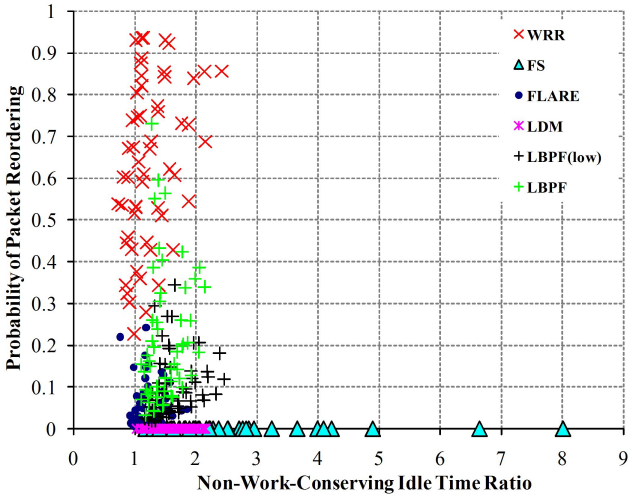


Fig. 9. Bandwidth utilization vs. packet reordering

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). These can be demonstrated by the figures: the closer the points are to the y-axis, the farther they are from the x-axis, however, except FLARE. 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 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 a high packet-arrival rate, can maintain a low risk of packet reordering with increased load balancing deviation.

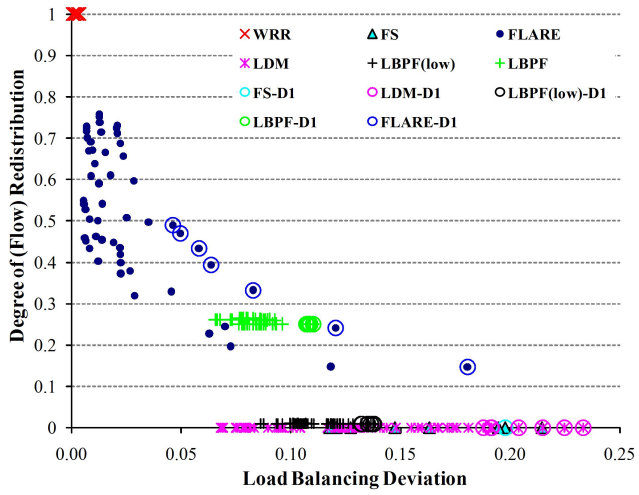


Fig. 10. Normalized degree of flow redistribution (due to load balancing) vs. load balancing efficiency

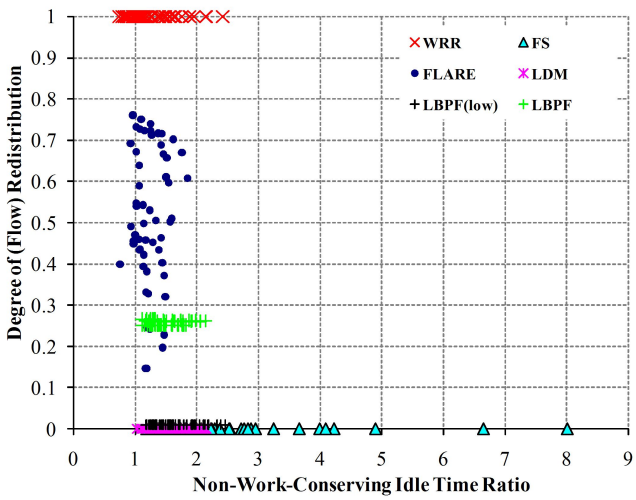


Fig. 11. Normalized degree of flow redistribution (due to load balancing) vs. bandwidth utilization efficiency

F. Degree of Flow Redistribution vs. Load Balancing and Bandwidth Utilization Efficiencies

Figs. 10 and 11 show normalized degrees of flow redistribution. 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.

Fig. 10 illustrates relations between the degree of flow redistribution and load balancing deviation. Obviously, the closer the points are to the y-axis, the farther they are from the x-axis. 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. 11 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.

VII. Concluding Remarks

As evidenced by several load balancing applications, exploitation of multiple communication paths is no longer only for single point of failure protection, but also for network provisioning. This article presents a comprehensive review 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. The performance of each model is evaluated by using different criteria, i.e., adaptability for dynamic traffic or network condition changes, load balancing and bandwidth utilization efficiencies, degree of flow redistribution, packet ordering preservation, communication overhead, computational complexity, and implementation 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. Their performance has also been demonstrated through simulations by using traffic traces observed in real networks.

References

[1] L. Golubchik, J. Lui, T. Tung, A. Chow, W. Lee, G. Franceschinis, and C. Anglano, "Multi-path continuous media streaming: what are the benefits?" *Perform. Eval.*, vol. 49, no. 1-4, pp. 429–449, Sep. 2002.

[2] J. Chen, W. Xu, S. He, Y. Sun, P. Thulasiraman, and X. Shen, "Utility-based asynchronous flow control algorithm for wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1116–1126, Sep. 2010.

[3] 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.

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

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

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

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

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

[9] Enhanced interior gateway routing protocol (EIGRP). Cisco white paper EIGRP. Cisco Systems Inc. [Online]. Available: http://www.cisco.com/warp/public/103/eigrp-toc.html

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

[11] 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.

[12] 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.

[13] M. Menth, A. Reifert, and J. Milbrandt, "Self-protecting multipaths - a simple and resource-efficient protection switching mechanism for MPLS networks," in *Proc. 3rd IFIP-TC6 Networking Conference (Networking)*, Athens, Greece, May 2004, pp. 526–537.

[14] M. Cheng, X. Gong, and L. Cai, "Joint routing and link rate allocation under bandwidth and energy constraints in sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3770–3379, Jul. 2009.

[15] 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.

[16] 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.

[17] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "Effective delay-controlled load distribution over multipath networks," *IEEE Trans. Parallel Distrib. Syst.*, 2011, vol. 22, no. 10, pp. 1730–1741, Oct. 2011.

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

[19] N. Wang, K. Ho, G. Pavlou, and M. Howarth, "An overview of routing optimization for internet traffic engineering," *IEEE Commun. Surveys Tutorials*, vol. 10, no. 1, pp. 36–56, 2008.

[20] I. Papapanagiotou, D. Toumpakaris, J. Lee, and M. Devetsikiotis, "A survey on next generation mobile WiMAX networks: objectives, features and technical challenges," *IEEE Commun. Surveys Tutorials*, vol. 11, no. 4, pp. 3–18, 2009.

[21] J. Duncanson and A. Berger, "Inverse multiplexing," *IEEE Commun. Mag.*, vol. 32, pp. 34–41, Apr. 1994.

[22] P. H. Fredette, "The past, present, and future of inverse multiplexing," *IEEE Commun. Mag.*, vol. 32, pp. 42–46, Apr. 1994.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for ipv4," RFC 4728, Feb. 2007.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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. Netw.*, vol. 17, no. 1, pp. 281–293, Feb. 2009.

[34] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on internet traffic identification," *IEEE Commun. Surveys Tutorials*, vol. 11, no. 3, pp. 37–52, 2009.

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

[36] 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.

[37] 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.

[38] DNS server round-robin functionality for cisco AS5800. Cisco Systems Inc. [Online]. Available: http://www.cisco.com/en/US/docs/ios/12_1t/12_1t3/feature/guide/dt_dnsrr.html

[39] Cisco localdirector configuration and command reference guide (software version 4.2.1). Cisco Systems Inc. [Online]. Available: http://www.cisco.com/en/US/docs/app_ntwk_services/waas/localdirector/command/v421/reference/LD42_ch03.html

[40] 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.

[41] 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. Netw.*, vol. 1, no. 3, pp. 344–357, Jun. 1993.

[42] How does unequal cost path load balancing (variance) work in IGRP and EIGRP? Cisco Systems Inc. [Online]. Available: http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a008009437d.shtml

[43] M. Lengyel, J. Sztrik, and C. S. Kim, "Simulation of differentiated services in network simulator," *Annales Universitatis Scientiarum Budapestinensis de Rolando Eőtvős Nominatae. Sectio Computatorica*, vol. 25, pp. 85–102, 2005.

[44] 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.

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

[46] K. C. Leung and V. O. K. Li, "Generalized load sharing for packet-switching networks: Theory and packet-based algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 7, pp. 694–702, Jul. 2006.

[47] A. Zinin, *Cisco IP routing: packet forwarding and intra-domain routing protocols*. Addison-Wesley, 2002.

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

[49] W. Shi, M. H. MacGregor, and P. Gburzynski, "Load balancing for parallel forwarding," *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 790–801, Aug. 2005.

[50] D. G. Thaler and C. V. Ravishankar, "Using name-based mappings to increase hit rates," *IEEE/ACM Trans. Netw.*, vol. 6, no. 1, pp. 1–14, Feb. 1998.

[51] 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.

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

[53] Y. Lee and Y. Choi, *An adaptive flow-level load control scheme for multipath forwarding*, Jul. 2001, vol. 2093.

[54] 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 Jose, CA, Oct. 2006, pp. 1–10.

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

[56] 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.

[57] 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.

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

[59] T. W. Chim, K. L. Yeung, and K.-S. Lui, "Traffic distribution over equal-cost-multi-paths," *Computer Networks*, vol. 49, no. 4, pp. 465–475, Nov. 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128605000411

[60] J. C. R. Bennett, C. Partridge, and N. Shectman, "Packet reordering is not pathological network behavior," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 789–798, Dec. 1999.
[61] 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.
[62] 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.
[63] 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.
[64] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," RFC 2330, May 1998.
[65] C. Demichelis and P. Chimento, "IP packet delay variation metric for IP performance metrics (IPPM)," RFC 3393, Nov. 2002.
[66] 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.
[67] F. Aune, *Cross-Layer Design Tutorial*. Trondheim, Norway: Norwegian University of Science and Technology, Nov. 2004, published under Creative Commons License.
[68] P. Danzig, J. Mogul, V. Paxson, and M. Schwartz. (1995, Mar.) The internet traffic archive. [Online]. Available: http://ita.ee.lbl.gov/index.html
[69] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, and J. Perser, "Packet reordering metrics," RFC 4737, Nov. 2006.
[70] A. Jayasumana, N. Piratla, T. Banka, and R. Whitner, "Improved packet reordering metrics," RFC 5236, Jun. 2008.

Sumet Prabhavat received the B.Eng. in Electrical Engineering from Chiangmai University, Thailand, in 1993, the M.Eng. in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2003, and the Ph.D. in Information Science from Tohoku University, Japan, in 2011. He joined KMITL's Research Center for Communications and Information Technology while studying for his M.Eng. degree. After graduation, he has been appointed as a Lecturer in the Faculty of Information Technology at the same university. His main research interests include load distribution, load balancing, performance analysis, application of queueing theory, and congestion control on communication networks. He received the Best Paper Award at the IEEE International Conference on Network Infrastructure and Digital Content (IEEE IC-NIDC) in 2009. He is an IEEE member.

Hiroki Nishiyama received his M.S. and Ph.D. in Information Science from Tohoku University, Japan, in 2007 and 2008, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science (JSPS) until finishing his Ph.D, when he then went on to become an Assistant Professor at the Graduate School of Information Sciences at Tohoku University. He has received Best Paper Awards from the IEEE Global Communications Conference 2010 (GLOBECOM 2010) as well as the 2009 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC 2009). He was also a recipient of the 2009 FUNAI Foundation's Research Incentive Award for Information Technology. His active areas of research include, traffic engineering, congestion control, satellite communications, ad hoc and sensor networks, and network security. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and an IEEE member.

Nirwan Ansari received the B.S.E.E. (*summa cum laude* with a perfect gpa) from the New Jersey Institute of Technology (NJIT), Newark, in 1982, the M.S.E.E. degree from University of Michigan, Ann Arbor, in 1983, and the Ph.D. degree from Purdue University, West Lafayette, IN, in 1988.

He joined NJITs Department of Electrical and Computer Engineering as Assistant Professor in 1988, tenured Associate Professor in 1993, and has been Full Professor since 1997. He has also assumed various administrative positions at NJIT. He authored *Computational Intelligence for Optimization* (New York: Springer, 1997, translated into Chinese in 2000) with E.S.H. Hou and edited *Neural Networks in Telecommunications* (New York: Springer, 1994) with B. Yuhas. His current research focuses on various aspects of broadband networks and multimedia communications. He has also contributed over 350 technical papers, over one third of which are in widely cited refereed journals/magazines.

He was/is serving on the Advisory Board and Editorial Board of eight journals, including as a Senior Technical Editor of *IEEE Communications Magazine* (2006-2009). He has been serving the IEEE in various capacities such as Chair of IEEE North Jersey COMSOC Chapter, Chair of IEEE North Jersey Section, Member of IEEE Region 1 Board of Governors, Chair of IEEE COMSOC Networking TC Cluster, Chair of IEEE COMSOC Technical Committee on Ad Hoc and Sensor Networks, and Chair/TPC Chair of several conferences/symposia. He has been frequently invited to deliver keynote addresses, distinguished lectures, tutorials, and talks. Some of his recent recognitions include an IEEE Fellow (Communications Society, Class of 2009), IEEE Leadership Award (2007, from Central Jersey/Princeton Section), the NJIT Excellence in Teaching in Outstanding Professional Development (2008), IEEE MGA Leadership Award (2008), the NCE Excellence in Teaching Award (2009), a couple of best paper awards, Thomas Alva Edison Patent Award, and designation as an IEEE Communications Society Distinguished Lecturer (2006-2009, two terms).

Nei Kato received his M.S. and Ph.D. Degrees in information engineering from Tohoku University, Japan, in 1988 and 1991, respectively. He joined Computer Center of Tohoku University at 1991, and has been a full professor at the Graduate School of Information Sciences since 2003. He has been engaged in research on computer networking, wireless mobile communications, image processing and neural networks. He has published more than 200 papers in journals and peer-reviewed conference proceedings.

Nei Kato currently serves as the chair of IEEE Satellite and Space Communications TC, the secretary of IEEE Ad Hoc & Sensor Networks TC, the chair of IEICE Satellite Communications TC, a technical editor of IEEE Wireless Communications(2006∼), an editor of IEEE Transactions on Wireless Communications(2008∼), an associate editor of IEEE Transactions on Vehicular Technology(2009∼). He served as a co-guest-editor for IEEE Wireless Communications Magazine SI on "Wireless Communications for E-healthcare", a symposium co-chair of GLOBECOM'07, ICC'10, ICC'11, ChinaCom'08, ChinaCom'09, and WCNC2010-2011 TPC Vice Chair.

His awards include Minoru Ishida Foundation Research Encouragement Prize(2003), Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee(2005), the FUNAI information Science Award(2007), the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion(2008), the IEICE Network System Research Award(2009), and best paper awards from many prestigious international conferences such as IEEE GLOBECOM, IWCMC, etc.

Besides his academic activities, he also serves as a member on the expert committee of Telecommunications Council, the special commissioner of Telecommunications Business Dispute Settlement Commission, Ministry of Internal Affairs and Communications, Japan, and as the chairperson of ITU-R SG4, Japan. Nei Kato is a member of the Institute of Electronics, Information and Communication Engineers (IEICE) and a senior member of IEEE.