Queuing Delays in Randomized Load Balanced Networks

R. Prasad
Georgia Tech.
ravi@cc.gatech.edu

P. J. Winzer, S. C. Borst, M. K. Thottan
Bell Labs, Lucent Technologies
{winzer,sem,marinat}@research.bell-labs.com

***Abstract*—Valiant's concept of Randomized Load Balancing (RLB), also promoted under the name 'two-phase routing', has previously been shown to provide a cost-effective way of implementing overlay networks that are robust to dynamically changing demand patterns. RLB is accomplished in two steps; in the first step, traffic is randomly distributed across the network, and in the second step traffic is routed to the final destination. One of the benefits of RLB is that packets experience only a single stage of routing, thus reducing queueing delays associated with multi-hop architectures. In this paper, we study the queuing performance of RLB, both through analytical methods and packet-level simulations using *ns2* on three representative carrier networks. We show that purely random traffic splitting in the randomization step of RLB leads to higher queuing delays than pseudo-random splitting using, e.g., a round-robin schedule. Furthermore, we show that, for pseudo-random scheduling, queuing delays depend significantly on the degree of uniformity of the offered demand patterns, with uniform demand matrices representing a provably worst-case scenario. These results are independent of whether RLB employs priority mechanisms between traffic from step one over step two. A comparison with multi-hop shortest-path routing reveals that RLB eliminates the occurrence of demand-specific hot spots in the network.**

I. Introduction

Emerging data communication services as well as content distribution and file sharing applications create an increasing amount of uncertainty and dynamism in the traffic distribution across carrier networks. Examples of such services are virtual private networks (VPNs), peer-to-peer networking, or remote storage and computing applications [1]. These services are well captured by the *hose model* [4], [5]. The hose model only specifies the node ingress/egress capacities, but does not specify the actual point-to-point demands. Thus it is up to the service provider to determine efficient routing and distribution of traffic within the network.

The traditional approach to routing and traffic distribution relies heavily on the accurate estimation of the traffic matrix. Accurate estimation is essential to avoid network congestion and to guarantee Quality of Service (QoS). Traffic matrix estimation requires fine-grained traffic monitoring which does not scale. When used for fine grained measurements, traffic monitoring based on the widely used Simple Network Management Protocol (SNMP) will significantly impact router performance. Therefore to account for the uncertainty of actual traffic demands, service providers often over-provision their networks.

One of the key requirements for networks to support emerging data services is the ability to handle extreme traffic variability and deal with hose constrained demand specifications. Ideally, to support a hose constrained traffic demand, a routing strategy must (*i*) be robust under the hose constraint; (*ii*) route traffic without creating hot spots; (*iii*) avoid the need for real time reconfiguration of capacity. All of these criteria are satisfied by the *Randomized Load Balancing* (RLB) scheme also known as *Two-Phase Routing*.

The basic idea of RLB is to route demands from network edge nodes in *two steps*. In the first (load balancing) step, all nodes randomly distribute their traffic among all nodes (or among a carefully chosen subset of nodes [19]) in the network. Traffic splitting may be performed on a packet-level or flow-level, and may be either done on layer 3 (IP) or layer 2 (Ethernet). In the second (routing) step, each node processes the traffic it received in step 1, and sends it to its final destination. Both steps of RLB carry traffic on statically pre-configured *circuits* or *paths*[1]. Due to the traffic randomization in step 1 of RLB, the architecture can handle extreme traffic variability. Hence, no reconfiguration is required to address dynamic changes. Furthermore, since each packet is only processed *once* between source and destination, RLB reduces the need for multiple packet buffering, thus providing improved QoS, especially in terms of delay jitter.

A. Related Work

The RLB architecture was first proposed by Valiant in the context of processor interconnection networks [21]. This concept was then extended to the design of scalable switches [17]. The scheme has recently received further attention for architecting high-capacity internet packet routers [3], [7], [22]. More recently RLB has been proposed at a *network level* as an efficient way of designing backbone networks [10], [18], [23]. Reference 10 describes algorithms for optimizing RLB link resources in capacitated networks that allow fractional (multipath) routing. Reference 23 is primarily focused on measuring the effects of RLB on minimizing the fanout of routers at the edge. RLB has also received attention to address the challenges posed by new network applications such as file

[1]When referring to a 'circuit' or a 'path', we mean a logic connection between two end nodes that does not require any packet processing en-route. Such an object may be implemented, e.g., using SONET or WDM technology. More generally, it may also be implemented using MPLS tunnels; however, the latter do not constitute 'circuits' in the strict sense, since (unlike SONET) MPLS requires packet label look-ups and buffering at each transit node. A more detailed account on this topic can be found in [19].

sharing and the Internet Indirection Infrastructure (i3). Reference 12 provides a linear program that computes the paths for maximum throughput to support highly variable service overlay traffic. References 11 and 19 show that compared to other routing strategies RLB requires less network resources.

B. Our Contribution

All of the previous work has proven the benefits of RLB by considering *time-averaged* capacities required for transport and switching in the network. In this paper we study the performance of RLB on a *packet level*. In particular, we investigate the queueing performance of RLB, both through analytical methods and through packet-level simulations using *ns2*. We show that purely probabilistic traffic splitting in the randomization step of RLB [24] leads to higher queuing delays than pseudo-random splitting using, e.g., a round-robin schedule. Furthermore, we show that, for pseudo-random scheduling, queuing delays depend significantly on the degree of uniformity of the offered demand patterns, with *uniform* demand matrices representing a provably *worst-case* scenario. These results are independent of whether one implements priority mechanisms between traffic from step 1 over traffic from step 2 in RLB. A comparison with packet-switched (multi-hop) architectures based on shortest-path routing reveals that RLB eliminates the occurrence of demand-specific hot spots in the network.

This paper is organized as follows: Section II reviews RLB in some more detail and outlines the architectural choices regarding queueing. Sec. III then presents a queuing analysis of RLB for various queueing options. In Sec. IV we describe the packet-level simulation and the underlying traffic model, followed by a discussion of results in Sec. V. Finally, Sec. VI summarizes the most important findings of this paper.

II. Randomized load balancing (RLB)

Figure 1 visualizes RLB from a queuing perspective for a four-node network. Each node consists of a local packet routing engine and an interface to a full mesh of paths.

In **step 1 of RLB** [Fig. 1(a)], the ingress traffic D is split and delivered to the routing engines of all other nodes, with $1/N$-th being kept locally, assuming that each node has the same amount of ingress/egress traffic. The traffic distribution in step 1 is random in the sense that it is totally agnostic of the demand matrix and does not require any routing decisions at the ingress. We will consider two different implementations of random traffic splitting in our analyses and simulations below, one based on *probabilistic* traffic splitting and the other based on a *pseudo-random* (e.g., round-robin) schedule. Generalizing to different traffic marginals D_i at each node, the amount of traffic to be distributed in step 1 of RLB is the product multicommodity flow [14] induced by the D_i's, i.e., the capacity required for the link between nodes i and j is $D_i D_j / \sum_i D_i$. Furthermore, it has been shown in Ref. 19 that load-balancing across a carefully chosen subset of $K < N$ intermediate nodes can provide cost and performance advantages; in this case of *selective* RLB, the ingress traffic would be split only among K nodes. Traffic splitting may be performed on a packet-by-packet basis or on a per-flow basis, and may be done on layer 3 (IP) or on layer 2 (Ethernet).

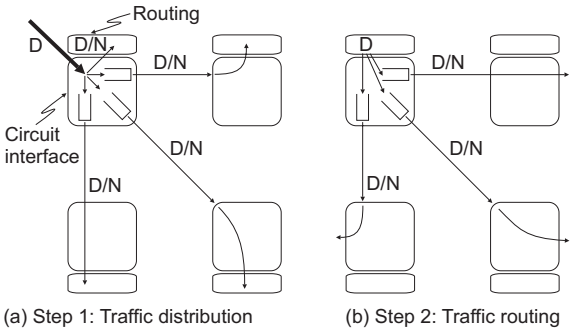


Fig. 1. Basic RLB architecture. Each node consists of a small routing engine, capable of routing an amount of traffic corresponding to the respective node's ingress/egress demand D, as well as an interface to the underlying circuit network, making up a full mesh of static circuits with capacity D/N.

In **step 2 of RLB** [Fig. 1(b)], a total traffic of D (with D/N stemming from each node in the network) is processed at each node's routing engine, and is statistically multiplexed on a path leading to its final destination, which, like in step 1, has capacity D/N for equal node ingress/egress capacities and $D_i D_j / \sum_i D_i$ for different capacities. Note that the traffic in steps 1 and 2 is uniform on average, *regardless* of the actual demand matrix to be routed, with fluctuations being accommodated by appropriate buffering within the routing nodes, as will be quantified later in this paper. The uniform nature of the traffic in each step of RLB permits pre-allocation of *static* network capacity, which dramatically simplifies network reliability and design.

Since RLB performs strict double-hop routing, all traffic is *buffered only once* (at the beginning of step 2). This reduces random buffering delays when compared to a multi-hop network architecture, which buffers traffic at each node. Furthermore, since *through-traffic* is not processed multiple times on a packet level on its way from source to destination, the network scalability problem associated with the difficulties in building large packet routers [7] is ameliorated by RLB.

One obvious disadvantage of RLB (as with any other architecture employing multi-path routing) is the routing of traffic over paths with significant time-of-flight differences. If traffic splitting is done on a packet level rather than per flow, the resulting delay spread can lead to packet mis-sequencing which potentially asks for packet re-ordering. Note, however, that these time-of-flight differences do *not* contribute to random delay jitter, but are fully *predictable* based on easily accessible knowledge of the two paths used by a packet on its way from source to destination. Therefore, these propagation delay differences can be counteracted by deterministic delays at the ingress, intermediate, or egress nodes, similar to what is being done when using virtual concatenation (VCAT) over multiple parallel routes in SONET. Recently in this context [15], a novel contention resolution mechanism was presented that enforces packet ordering in a load balanced switch ar-

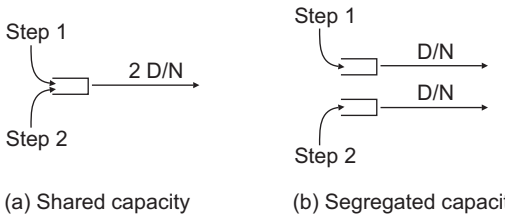


Fig. 2. In RLB, a link can be shared among step 1 and step 2 traffic, or capacities can be segregated.

chitecture. The maximum propagation delay in RLB is about *twice* the propagation delay of the longest path in the network, which depending on the underlying application may restrict the geographic dimensions of such networks.

III. Queuing analysis

We now analyze the queueing dynamics for both the random and the pseudo-random traffic splitting schemes. We consider the queues associated with step 1 (traffic splitting) as well as step 2 (routing). We will distinguish between two scenarios, depending on whether the two queues share the total link bandwidth of $2D/N$ in a work-conserving manner [Fig. 2(a)], or receive dedicated portions of D/N each [Fig. 2(b)]. For convenience, we assume in the analysis that the packets have a fixed size and that the system operates in a time-slotted fashion, with the duration of a time slot equal to a packet service time.

A. Pseudo-random traffic splitting

We first examine the pseudo-random traffic splitting scheme. In preparation for the queueing analysis, we start with determining the statistical characteristics of the packet arrival processes on the link from node k to node j. Let $\alpha_{ij} = (1-\epsilon)d_{ij}$ be the probability that during an arbitrary time slot a packet arrives at (ingress) node i destined for (egress) node j, with $d_{ij} = D_{ij}/D$. Denote $\alpha_j = \sum_{i=1}^{N} \alpha_{ij}$.

By virtue of the traffic splitting scheme, the arrival pattern of the step-1 traffic is statistically identical for all j. Hence, we drop the index j, and let B_k be a random variable representing the number of packet arrivals of the step-1 traffic per time slot. Note that B_k is simply a 0–1 random variable with $\mathbb{P}\{B_k = 1\} = \alpha_k$ and thus $\mathbb{P}\{B_k = 0\} = 1 - \alpha_k$.

By construction, the arrival pattern of the step-2 traffic is statistically identical for all k. Hence, we suppress the index k, and let A_j be a random variable representing the number of packet arrivals of the step-2 traffic per time slot. Then A_j may be represented as $A_j = \sum_{i=1}^{N} A_{ij}$, where the A_{ij}'s are independent 0–1 random variables with $\mathbb{P}\{A_{ij} = 1\} = \alpha_{ij}$ and thus $\mathbb{P}\{A_{ij} = 0\} = 1 - \alpha_{ij}$. Denote by $\alpha_j^{(m)} := \mathbb{E}\{A_j^m\}$ the m-th moment of A_j. Note that $\alpha_j^{(1)} = \sum_{i=1}^{N} \alpha_{ij} = \alpha_j$,

$$\alpha_j^{(2)} = \alpha_j(\alpha_j + 1) - \sum_{i=1}^{N} \alpha_{ij}^2, \tag{1}$$

and

$$\begin{aligned} \alpha_j^{(3)} &= \alpha_j(\alpha_j^2 + 3\alpha_j + 1) - 3(\alpha_j + 1)\sum_{i=1}^{N} \alpha_{ij}^2 + 2\sum_{i=1}^{N} \alpha_{ij}^3 \\ &= -\alpha_j(2\alpha_j^2 + 3\alpha_j + 2) + 3(\alpha_j + 1)\alpha_j^{(2)} + 2\sum_{i=1}^{N} \alpha_{ij}^3. \end{aligned}$$

It is easily verified from convexity arguments that, for a fixed mean α_j, the values of $\alpha_j^{(2)}$, $\alpha_j^{(3)}$ are minimal for binary matrices $\{d_{ij}\}$ and maximal for uniform matrices $\{d_{ij}\}$, i.e., $d_{ij} = 1/(N-1)$ (assuming $d_{jj} \equiv 0$), yielding

$$\alpha_j \le \alpha_j^{(2)} \le \alpha_j(\alpha_j + 1) - \frac{\alpha_j^2}{N-1} \le \alpha_j(1 + \alpha_j), \tag{2}$$

$$\begin{aligned} \alpha_j &\le \alpha_j^{(3)} && (3) \\ &\le \alpha_j(\alpha_j^2 + 3\alpha_j + 1) - 3(\alpha_j + 1)\frac{\alpha_j^2}{N-1} + 2\frac{\alpha_j^3}{(N-1)^2} \\ &\le \alpha_j(\alpha_j^2 + 3\alpha_j + 1). && (4) \end{aligned}$$

The above upper bounds for $\alpha_j^{(2)}$, $\alpha_j^{(3)}$ correspond to the second and third moment of a Poisson distributed random variable $\tilde{A}_j$ with mean α_j, and will be quite tight when N is not too small and the α_{ij}'s are close to uniform.

1) Segregated bandwidth: Armed with a statistical description of the packet arrival processes, we are now in a position to analyze the queueing behavior. Let $Q_{k,1}(t)$ and $Q_{j,2}(t)$ be the queue sizes at the start of the t-th time slot for the step-1 (link to intermediate node k) traffic and the step-2 (link to egress node j) traffic, respectively. We first investigate the scenario where the total link bandwidth is statically partitioned between the two queues. Observe that $Q_{k,1}(t)$ will be identically zero, since no more than one packet arrives per time slot in the pseudo-random case [Fig. 2(b)]. Let $A_j(t)$ be the number of packet arrivals between the start of the t-th and $(t+1)$-th time slot. The evolution of the process $Q_{j,2}(t)$ over time may be described by the simple recursion

$$Q_{j,2}(t+1) = [Q_{j,2}(t) + A_j(t) - 1]^+, \tag{5}$$

with the notational convention $[q]^+ := \max\{0, q\}$. If the destinations of arriving packets at the ingress nodes in successive time slots are assumed to be uncorrelated, then the $A_j(t)$'s are independent and identically distributed (i.i.d.) copies of the random variable A_j. Let $A_j(z) := \mathbb{E}\{z^{A_j}\}$ and $Q_j(z) := \mathbb{E}\{z^{Q_{j,2}}\}$ be the probability generating functions (pgf's) of the random variables A_j and $Q_{j,2}$, respectively, with $Q_{j,2}$ representing the steady-state version of $Q_{j,2}(t)$. It may then be derived from the recursion (5),

$$Q_{j,2}(z) = \frac{(1-\alpha_j)(1-z)}{A_j(z) - z},$$

which gives

$$\mathbb{E}\{Q_{j,2}\} = \frac{\alpha_j^{(2)} - \alpha_j}{2(1-\alpha_j)}, \quad (6)$$

$$\mathbb{E}\{Q_{j,2}^2\} = 2(\mathbb{E}\{Q_{j,2}\})^2 + \mathbb{E}\{Q_{j,2}\} + R, \quad (7)$$

$$stdev(Q_{j,2}) = \sqrt{(\mathbb{E}\{Q_{j,2}\})^2 + \mathbb{E}\{Q_{j,2}\} + R}, \quad (8)$$

with

$$R := \frac{\alpha_j^{(3)} - 3\alpha_j^{(2)} + 2\alpha_j}{3(1-\alpha_j)}.$$

Using the upper bounds for $\alpha_j^{(2)}$, $\alpha_j^{(3)}$ in equations (2), (4), we obtain

$$0 \leq \mathbb{E}\{Q_{j,2}\} \leq \frac{\alpha_j^2}{2(1-\alpha_j)},$$

where the lower bound is attained for binary matrices $\{d_{ij}\}$ and the upper bound, corresponding to a Poisson arrival process $\tilde{A}_j(t)$, is approached for uniform matrices $\{d_{ij}\}$ when N is not too small.

2) Intermezzo: Before moving to the case of shared bandwidth, we first pause to make some important observations.

Stochastic majorization properties

The above results indicate that the mean queue size is minimal for binary matrices $\{d_{ij}\}$ and maximal for uniform matrices $\{d_{ij}\}$. This result is in fact an implication of a far more general property, namely that the queue size Q_j is 'larger' when the vector $(\alpha_{1j}, \ldots, \alpha_{Nj})$ is 'more balanced'. In order to formalize the above statement, we need to introduce some technical concepts. For a given vector $\alpha \in U^N := [0,1]^N$, define the associated random variable $A_\alpha := \sum_{i=1}^N A_i$, with $\mathbb{P}\{A_i = 1\} = \alpha_i$ and $\mathbb{P}\{A_i = 0\} = 1 - \alpha_i$. Also, for a given random variable A, define the process $Q_A(t)$ by the recursion $Q_A(t) := [Q_A(t-1) + A(t-1) - 1]^+$. The next proposition states that $Q_{A_{\alpha'}}$ is larger than $Q_{A_{\alpha''}}$ in the increasing convex ordering sense when the vector α' majorizes the vector α''. (For definitions of these concepts, see Refs. 16 and 20.)

Proposition 1 If $\alpha' \prec \alpha''$, then $Q_{A_{\alpha'}}(t) \geq_{\rm icx} Q_{A_{\alpha''}}(t)$ for all $t = 1, 2, \ldots$, $Q_{A_{\alpha'}} \geq_{\rm icx} Q_{A_{\alpha''}}$, and $\mathbb{E}\{Q^m_{A_{\alpha'}}\} \geq \mathbb{E}\{Q^m_{A_{\alpha''}}\}$ for all $m = 1, 2, \ldots$. In particular, if $\alpha = a(1, 1, \ldots, 1)$ then $Q_{A_\alpha} \geq_{\rm icx} Q_{A_\beta}$ for any vector $\beta \in U^N$ with $\sum_{i=1}^N \beta_i/N = a < 1$.

Observe that $\alpha^{(K)} = (1/K, 1/K, \ldots, 1/K) \sum_{i=1}^N \alpha_i \prec \alpha$ for any N-dimensional vector α, $N \leq K$. In addition, it may be shown that the random variable $A^{(K)}$ converges to a Poisson distributed random variable $\tilde{A}_j$ with mean $\sum_{i=1}^N \alpha_i$ as $K \to \infty$. These two observations imply that, informally speaking, the queueing behavior is guaranteed to be 'better than Poisson', which is formalized in the next corollary.

Corollary 1 For any vector α, $Q_{A_\alpha} \leq_{\rm icx} Q_{\tilde{A}}$.

Heavy-traffic behavior

The above results also show that $stdev(Q_{j,2})/\mathbb{E}\{Q_{j,2}\} \to 1$ as $\epsilon \downarrow 0$, i.e., $\alpha_j \uparrow 1$. Moreover, for given relative fractions $\alpha_{ij} = (1-\epsilon)d_{ij}$, the mean queue size $\mathbb{E}\{Q_{j,2}\}$ approximately grows linearly with $1/\epsilon$. These two results are again consequences of a more general property, namely that the scaled queue size $\epsilon Q_{j,2}$ converges to an exponentially distributed random variable with mean $(1 - d_j^{(2)})/2$ as $\epsilon \downarrow 0$, i.e.,

$$\lim_{\epsilon \downarrow 0} \mathbb{P}\{\epsilon Q_{j,2} > x\} = {\rm e}^{-2x/(1-d_j^{(2)})},$$

with $d_j^{(2)} := \sum_{i=1}^N d_{ij}^2$. Note that $1 - d_j^{(2)}$ arises as the limit of $\alpha_j^{(2)} - \alpha_j$ as $\alpha_j \uparrow 1$ for given relative fractions $\alpha_{ij} = (1-\epsilon)d_{ij}$. This suggests the following approximation for the queue size distribution:

$$\mathbb{P}\{Q_{j,2} > x\} \approx {\rm e}^{-2(1-\alpha_j)x/(\alpha_j^{(2)} - \alpha_j)} = {\rm e}^{-x/\mathbb{E}\{Q_{j,2}\}} = \sigma_j^x,$$

with $-1/\mathbb{E}\{Q_{j,2}\}$ the asymptotic decay exponent and $\sigma_j := {\rm e}^{-1/\mathbb{E}\{Q_{j,2}\}}$ the asymptotic decay factor.

3) Shared bandwidth: We now proceed to the scenario where the total link bandwidth is dynamically shared between the two queues in a work-conserving manner [Fig. 2(a)]. Let $Q_{jk}(t) := Q_{k,1} + Q_{j,2}(t)$ be the total queue size at the start of the t-th time slot. The evolution of the process $Q_{jk}(t)$ over time may be described by the recursion

$$Q_{jk}(t+1) = [Q_{jk}(t) + A_j(t) + B_k(t) - 2]^+, \quad (9)$$

with $A_j(t)$ and $B_k(t)$ i.i.d. copies of the random variables A_j and B_k. Let $Q_{jk}(z) := \mathbb{E}\{z^{Q_{jk}}\}$ be the pgf of Q_{jk}, with Q_{jk} representing the steady-state version of $Q_{jk}(t)$. Then it may be derived from the recursion (9),

$$Q_{jk}(z) = \frac{(q_0 + (q_0 + q_1)z)(1-z)}{A_j(z)B_k(z) - z^2},$$

with $q_m := \mathbb{P}\{Q_{jk} = m\}$, $m = 0, 1$. The probabilities q_0, q_1 can be expressed in terms of the roots of certain equations, but can generally not be expressed in closed form. However, there are simple sharp lower and upper bounds:

$$\frac{\alpha_j^{(2)} - \alpha_j^2 - \alpha_k^2 + \alpha_k}{4(1-\alpha_{j+k})} - \alpha_{j+k} \leq \mathbb{E}\{Q_{jk}\} \leq \frac{\alpha_j^{(2)} - \alpha_j^2 - \alpha_k^2 + \alpha_k}{4(1-\alpha_{j+k})},$$

with $\alpha_{j+k} := (\alpha_j + \alpha_k)/2$. Assuming $\alpha_j = \alpha_k = \alpha$, we obtain

$$\frac{\alpha_j^{(2)} - \alpha}{4(1-\alpha)} - \alpha/2 \leq \mathbb{E}\{Q_{jk}\} \leq \frac{\alpha_j^{(2)} - \alpha}{4(1-\alpha)} + \alpha/2. \quad (10)$$

Note that **the dominant term is exactly half the mean queue size $\mathbb{E}\{Q_{j,2}\}$ in the case of segregated bandwidth, and hence $\mathbb{E}\{Q_{jk}\}/\mathbb{E}\{Q_{j,2}\} \to 1/2$ as $\alpha \uparrow 1$.**

The behavior of the two individual queue sizes depends on precisely how the total bandwidth is shared. However, as long as the step-1 traffic is guaranteed to receive at least half of the total bandwidth, $Q_{k,1}(t)$ continues to be identically zero, which means $Q_{j,2}(t) \equiv Q_{jk}(t)$.

B. Random traffic splitting

We now turn the attention to the purely random traffic splitting scheme, and will indicate the variables with a hat to distinguish them from those used in the pseudo-random case. As before, we first analyze the statistical characteristics of the packet arrival processes on the link from node k to node j before proceeding to the queueing analysis.

Let $\hat{B}_k$ be a random variable representing the number of packet arrivals of the step-1 traffic per time slot. Then $\hat{B}_k$ may be represented as

$$\hat{B}_k = \sum_{l=1}^{N} \hat{B}_{kl},$$

where the $\hat{B}_{kl}$ are independent 0–1 random variables with $\mathbb{P}\{\hat{B}_{kl} = 1\} = \alpha_k/N$ and thus $\mathbb{P}\{\hat{B}_{kl} = 0\} = 1 - \alpha_k/N$. Denote by $\hat{\beta}_k^{(m)} = \mathbb{E}\{\hat{B}_k^m\}$ the m-th moment of $\hat{B}_k$. Note that $\hat{\beta}_k^{(1)} = \alpha_k$. One can also calculate the higher moments $\hat{\beta}_k^{(m)}$ in a similar fashion as before. However, the more relevant observation is that $\hat{B}_k$ is again smaller (in the convex ordering sense) than a Poisson distributed random variable $\tilde{A}_k$ with mean α_k. The latter upper bound will be quite tight as long as N is not too small, *regardless* of the degree of (non)-uniformity of the α_{ij}'s.

Turning to the step-2 traffic now, let $\hat{A}_j$ be a random variable representing the number of packet 'arrivals' per time slot. The word 'arrivals' is put in quotes, because not all of these packets may actually make it to step 2 right away due to possible queueing at step 1. The variable $\hat{A}_j$ may be represented as

$$\hat{A}_j = \sum_{i=1}^{N} \sum_{l=1}^{N} \hat{A}_{ijl},$$

where the $\hat{A}_{ijl}$ are independent 0–1 random variables with $\mathbb{P}\{\hat{A}_{ijl} = 1\} = \alpha_{ij}/N$ and thus $\mathbb{P}\{\hat{A}_{ijl} = 0\} = 1 - \alpha_{ij}/N$. Denote by $\hat{\alpha}_j^{(m)} = \mathbb{E}\{\hat{A}_j^m\}$ the m-th moment of $\hat{A}_j$. Note that $\hat{\alpha}_j^{(1)} = \alpha_j$. One can also calculate the higher moments $\hat{\alpha}_j^{(m)}$ in a similar fashion as before. However, the more relevant observation is that $\hat{A}_j$ is again smaller (in the convex ordering sense) than a Poisson distributed random variable $\tilde{A}_j$ with mean α_j. The latter upper bound will be quite tight as long as N is not too small, even when the α_{ij}'s are far from uniform.

1) Segregated bandwidth: Having obtained a characterization of the packet arrival processess, we now analyze the queueing behavior. We first examine the scenario where the total link bandwidth is partitioned between the two queues [Fig. 2(b)]. From the recursion $\hat{Q}_{k,1}(t+1) = [\hat{Q}_{k,1}(t) + \hat{B}_k(t) - 1]^+$, it can be shown that

$$\mathbb{E}\{z^{\hat{Q}_{k,1}}\} = \frac{(1-\alpha_k)(1-z)}{\mathbb{E}\{z^{\hat{B}_k}\} - z},$$

and one can calculate the moments of $\hat{Q}_{k,1}$ as done before for $Q_{j,2}$ with $\alpha_j^{(m)}$ replaced by $\hat{\beta}_k^{(m)}$. Moreover, it can be shown that $\hat{Q}_{k,1}$ is smaller in the increasing convex ordering sense than $\tilde{Q}_{k,1}$, with

$$\mathbb{E}\{z^{\tilde{Q}_{k,1}}\} = \frac{(1-\alpha_k)(1-z)}{\mathbb{E}\{z^{\tilde{A}}\} - z},$$

with $\mathbb{E}\{z^{\tilde{A}_k(z)}\} = \mathrm{e}^{-\alpha_k(1-z)}$, so that

$$\mathbb{E}\{\tilde{Q}_{k,1}\} \le \mathbb{E}\{\tilde{Q}_{k,1}\} = \frac{\alpha_k^2}{2(1-\alpha_k)},$$

$$\mathbb{E}\{\tilde{Q}_{k,1}^2\} \le \mathbb{E}\{\tilde{Q}_{k,1}^2\} = \frac{\alpha_k^3}{3(1-\alpha_k)} + \mathbb{E}\{\tilde{Q}_{k,1}\} + 2(\mathbb{E}\{\tilde{Q}_{k,1}\})^2.$$

The above upper bounds will be quite tight when N is not too small.

It is difficult to determine the distribution of $\hat{Q}_{j,2}$. However, it can be shown that $\hat{Q}_{j,2}$ is larger than the queue size $Q_{j,2}$ in the pseudo-random case.

2) Shared bandwidth: Finally we investigate the scenario where the total link bandwidth is dynamically shared between the two queues in a work-conserving manner [Fig. 2(a)]. Let $\tilde{Q}_{jk}$ be the size of a queue fed by a Poisson arrival process $\tilde{A}_{j+k}(t)$ with mean $\alpha_{j+k} = (\alpha_j + \alpha_k)/2$. Assuming $\alpha_j = \alpha_k = \alpha$, it can be shown that $\hat{Q}_{jk}$ is smaller (in the increasing convex ordering sense) than $\tilde{Q}_{jk}$, with

$$\mathbb{E}\{z^{\tilde{Q}_{jk}}\} = \frac{(1-\alpha)(1-z)}{\mathbb{E}\{z^{\tilde{A}_{j+k}}\} - z},$$

with $\mathbb{E}\{z^{\tilde{A}_{j+k}}\} = \mathrm{e}^{-\alpha(1-z)}$, so that

$$\mathbb{E}\{\hat{Q}_{jk}\} \le \mathbb{E}\{\tilde{Q}_{jk}\} = \frac{\alpha^2}{2(1-\alpha)}, \tag{11}$$

$$\mathbb{E}\{\hat{Q}_{jk}^2\} \le \mathbb{E}\{\tilde{Q}_{jk}^2\} = \frac{\alpha^3}{3(1-\alpha)} + \mathbb{E}\{\tilde{Q}_{jk}\} + 2(\mathbb{E}\{\tilde{Q}_{jk}\})^2. \tag{12}$$

The above upper bounds will be quite tight when N is not too small. Comparing the above results with those for the case of segregated bandwidth, we conclude that **the total queue size $\hat{Q}_{jk}$ is now about as large as just the queue size $\hat{Q}_{k,1}$ of the step-1 traffic in the latter case**.

We conclude the section with an important remark. In the above queueing analysis we have assumed that the destinations of the arriving packets at the various ingress nodes are i.i.d. from slot to slot. Of course, there are a far broader range of arrival processes imaginable which satisfy the marginal statistics implied by the matrices $\{d_{ij}\}$, and in particular ones which exhibit strong temporal correlations. Correlations in the destinations of arriving packets render an exact analysis intractable in general, and a detailed treatment is beyond the scope of the present paper. However, it can be shown along similar lines as in Ref. 2 that under mild assumptions the random variables $A_j(t)$ converge to a sequence of independent Poisson distributed random variables $\tilde{A}_j(t)$ with mean α_j,

and the resulting queue sizes also converge to those of a queue fed by a Poisson arrival process as N grows large. We refer to Ref. 13 for related results in the context of RLB. This suggests that random traffic splitting schemes provide an effective mechanism for 'breaking' temporal correlations in the activity patterns of individual node pairs, in a similar way as the impact of correlations in the activity of individual sources diminishes at high levels of statistical multiplexing. It further provides justification for drawing on available queueing results for models with Poisson arrivals as approximations [17] for more intricate and bursty traffic processes.

IV. Packet-level simulations

A. Simulation model

We implemented RLB as a new application in *ns2* for packet-level simulations. In this application, each node marks the packets with the address of the destination node and sends them to an intermediate node. The intermediate nodes are selected with a round-robin scheme for investigating the behavior of pseudo-random schedules. To investigate the behavior of a fully random schedule, the intermediate nodes are selected by generating a uniform random integer between 1 and N. At every ingress node, the rate at which a particular node is selected as the destination is determined by randomly chosen traffic matrices satisfying the hose constraint (see Sec. IV-C). Upon receiving a packet, the intermediate node looks up the destination from the packet and forwards it to the destination node. Flows from step 1 and step 2 of RLB, as defined in Sec. II, are considered parts of two different traffic classes, and class-based queuing (CBQ) can be used to assign priorities to these classes.

For comparative purposes (Sec.V-D), a shortest path routing scheme was also implemented, using Dijkstra's algorithm based on the delay metric. The delay metric was obtained from the propagation delays on each of the links on the respective topologies.

B. Example networks

We use the three representative carrier networks depicted in Fig. 3 as examples to study the queuing performance of different network architectures: the UK research network Janet, the US research backbone Abilene, and the European research network Geant. The key network characteristics are summarized in Tab. I.

Also shown in Tab. I is the sum of all required link capacities in the three networks that guarantees routing of all possible hose matrices using shortest-path (SP) routing, VPN-Tree routing, and RLB. These capacities, taken from Ref. 19, are the results of linear programming (LP) formulations and are normalized to $D = 1$. In VPN-Tree routing, one determines that tree on the physical network topology that yields lowest total link capacities under the hose constraint and only assigns capacity to the links that are part of that tree. It has been shown [6] that VPN-Tree routing represents the optimum routing strategy for hose traffic in the sense that it uses the least amount of total link capacities. Note that (*i*) RLB always uses at least as much link capacity as VPN-Tree routing and (*ii*) SP routing in general uses more link capacity than RLB for the same degree of robustness to traffic pattern variations [19]. In this work we only focus on RLB and SP, since we assume that the set up, maintenance, and restoration costs associated with VPN-tree topologies are non-trivial and hence may outweigh their capacity advantages.

TABLE I
Main characteristics of our three example networks.

	Janet	Abilene	Geant
Number of nodes	8	11	27
Number of links	10	14	40
Average link distance [km]	184	1,317	797
Link capacities × km (SP)	3,437	37,019	69,142
Link capacities × km (VPN)	2,302	22,621	36,823
Link capacities × km (RLB)	2,776	30,087	56,312

C. Traffic model

To study the queuing performance of our three example networks, we used sets of randomly generated *hose traffic matrices* [5], [4]. Hose matrices are characterized by the fact that each node i has fixed ingress capacities $D_{i,\text{ingress}}$ and egress capacities $D_{i,\text{egress}}$, which is a reasonable assumption motivated by the physical connection speed attached to each node. The point-to-point demands d_{ij} of the hose matrices obey the ingress and egress relationships $\sum_j d_{ij} = D_{i,\text{ingress}}$ and $\sum_i d_{ij} = D_{j,\text{egress}}$, and there is no traffic from any node destined to itself, i.e., the matrices have zero diagonals. No effort is made to symmetrize the traffic matrices, i.e., $d_{ij} \neq d_{ji}$. However, we do assume all ingress and egress capacities to be equal for all nodes i, i.e., $D_{i,\text{ingress}} = D_{i,\text{egress}} = D$. In our packet-level simulations we further assume $D = 1$ Mbps, with fixed-size 1500 byte packets.

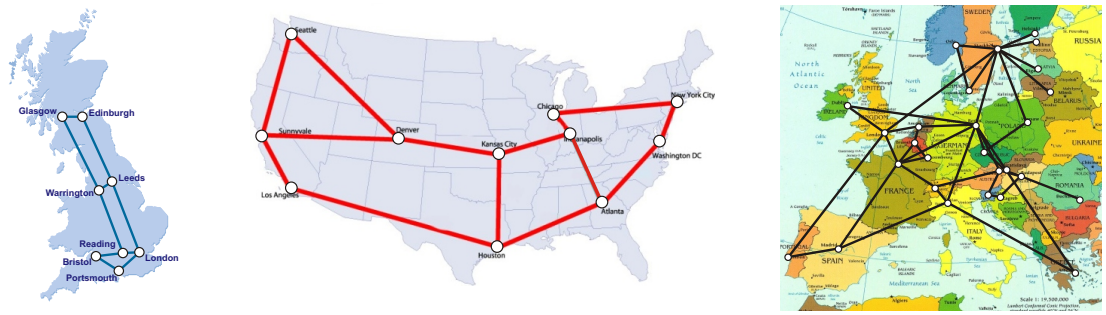
Motivated by the findings of our queuing analyses in Sec. III, in particular Eq. (1), we took the sum-of-squares of all matrix elements as our metric μ to describe different traffic matrices,

$$\mu = \sum_{i,j} d_{ij}^2 \ . \tag{13}$$

We randomly generated 10 hose matrices for each prescribed metric μ. In agreement with our findings in Sec. III, the smallest metric of $D^2N/(N-1)$ is obtained for *uniform* matrices with $d_{ij} = D/(N-1)$. Conversely, the largest metric of D^2N is obtained for random permutations (with non-zero diagonal) of the identity matrix. Figure 4 depicts the metrics μ for the random hose matrices generated for our 27-node (Geant), 11-node (Abilene), and 8-node (Janet) networks. Each horizontal slot (indicated by the dotted vertical lines) contains 10 randomly chosen hose matrices with roughly the same metrics μ.

V. Results and discussion

In this section, we discuss the results of our packet-level simulations capturing the queuing behavior of RLB on our three example networks; we compare these results to our analyses in Sec. III.



(a) JANET topology (b) ABILENE topology (c) GEANT topology

Fig. 3. Three example networks considered in this paper [http://www.ja.net; http://www.abilene.iu.edu; http://www.geant.net].

A. *Impact of step 1 traffic splitting on queue sizes*

In studying the queuing performance of RLB, we first note that due to the randomization process in step 1 of RLB, all queues are statistically identical. We verified this fact by comparing the statistical parameters of all individual queues on our example networks. Therefore, we looked at network-averaged queue parameters in the frame of our studies.

Figure 5 shows the network-averaged queue sizes[2] (red) as well as the queue size standard deviations (blue) as a function of time for the three networks under consideration. Each time slot indicated by the dotted vertical lines comprises *ns2*-simulations of 10 randomly different hose traffic matrices with similar metrics μ [cf. Fig. 4]. As time progresses, the matrices gradually change from uniform to highly skewed. Each traffic matrix is applied for 100 s and the queues are monitored every 10 s, which was verified to be large enough for the queues to reach their steady states. The offered network load is chosen $\varepsilon = 5\%$ below capacity, i.e. the links have capacity D/N and the ingress traffic is $0.95D$.

Figure 5(a) applies to the aggregate (step 1 plus step 2) queue of RLB with *probabilistic* step 1 traffic splitting on the Janet network. We observe that the **average queue size as well as its standard deviation are virtually independent of the applied traffic matrix**. This is expected from the complete traffic randomization process performed in step 1 and also agrees with our analyses in Sec. III.

[2]The queue sizes are measured in kB, with the understanding that 1 kB=1000 B.

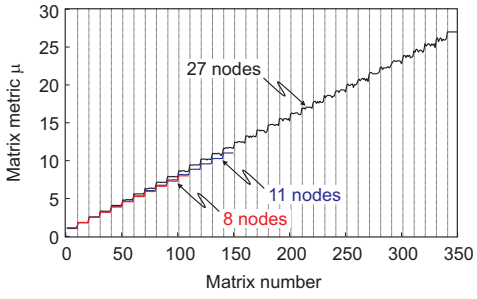


Fig. 4. Metrics μ of our randomly generated traffic matrices for the three example networks studied here.

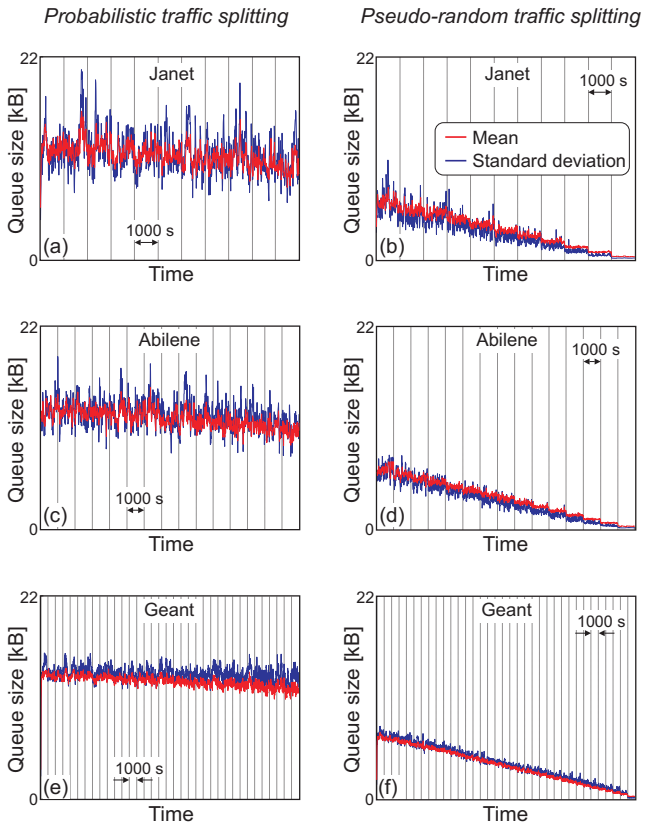


Fig. 5. Mean queue sizes (red) and standard deviations (blue) as a function of time for RLB using probabilistic step 1 traffic splitting (a,c,e) and pseudo-random splitting (b,d,f) on our example networks. As time progresses, the random test matrices become increasingly skewed (increasing metrics μ).

Fig. 5(b) applies to a *pseudo-random* traffic distribution on the Janet network, which in our case was implemented as a round-robin schedule. We again see close agreement between mean and standard deviation of the queues. Interestingly, the **average queue size takes on its worst-case values for uniform traffic matrices, while highly skewed matrices result in smaller queues**, in agreement with our predictions of Sec. III. This behavior can be intuitively understood from the fact that a deterministic step 1 schedule maximally smoothens the distribution of final packet destinations at an intermediate node if the ingress traffic at each node is destined for just

one other egress node. In this case, within a time duration corresponding to N packets, each intermediate node receives exactly one packet destined for each output node, which matches the allocated D/N link capacity for step 2 traffic and thus eliminates any queue build-up. Uniform traffic patterns, on the other hand, may lead to multiple packets out of N packets arriving at an intermediate node that are destined for one particular output node, which is larger than what the D/N-link can support and thus lets queues build up, similar to the probabilistic traffic splitting scenario.

By comparing Figs. 5(a) and (b), we note that the **worst-case queue size for the pseudo-random schedule is lower than the queue size for probabilistic traffic splitting**. This can be understood from the fact that a pseudo-random step 1 schedule assigns any given intermediate node to exactly 1 out of N packets upon step 1 traffic splitting, which matches the D/N link capacities for step 1 and therefore avoids any step 1 queue build-up and (for networks operating below capacity) frees up any unused step 1 link capacity for step 2 traffic, i.e., there is a service rate of $(1+\varepsilon)D/N$ available on each link for step 2 traffic, while the offered load is $(1-\varepsilon)D/N$. This is equivalent to an offered load of $(1-\varepsilon)/(1+\varepsilon)D/N \approx (1-2\varepsilon)D/N$ over a link of capacity D/N, as opposed to an offered load of $(1-\varepsilon)D/N$ over a link of capacity D/N for RLB using probabilistic traffic splitting.

We also observe that throughout our simulations the **standard deviation of the queue size equals its mean, which is indicative of an exponential queue size distribution**. The exponential nature of the queue size distribution (straight lines on a logarithmic scale) is indeed confirmed by numerically evaluating the cumulative queue size densities, shown in Fig. 6(a) for probabilistic traffic splitting and (b) for pseudo-random traffic splitting under different levels of offered load for the Janet network. In (b), the matrix metric μ is assumed mid-way between the minimum and maximum possible values.

Looking at the average queue size and its standard deviation for the Abilene and Geant networks [Figs. 5(c,d) and (e,f), respectively], we note that the **studied queueing parameters (means and standard deviations) do not depend on on the size of the network** under consideration.

B. Priority mechanisms between step 1 and step 2 queuing

Having understood the importance of the nature of step 1 traffic splitting in RLB, we investigate whether the queueing

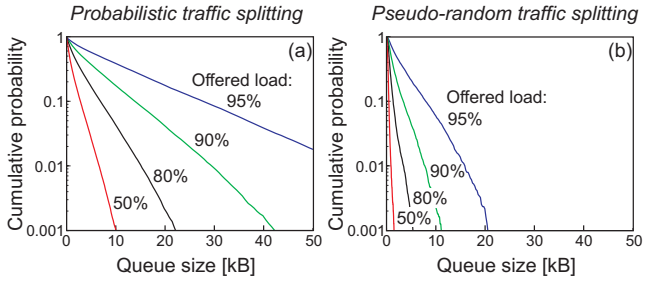


Fig. 6. Queue sizes are close to exponentially distributed for RLB with probabilistic (a) as well as with pseudo-random (b) traffic splitting. The curves apply to the Janet network with different levels of offered load.

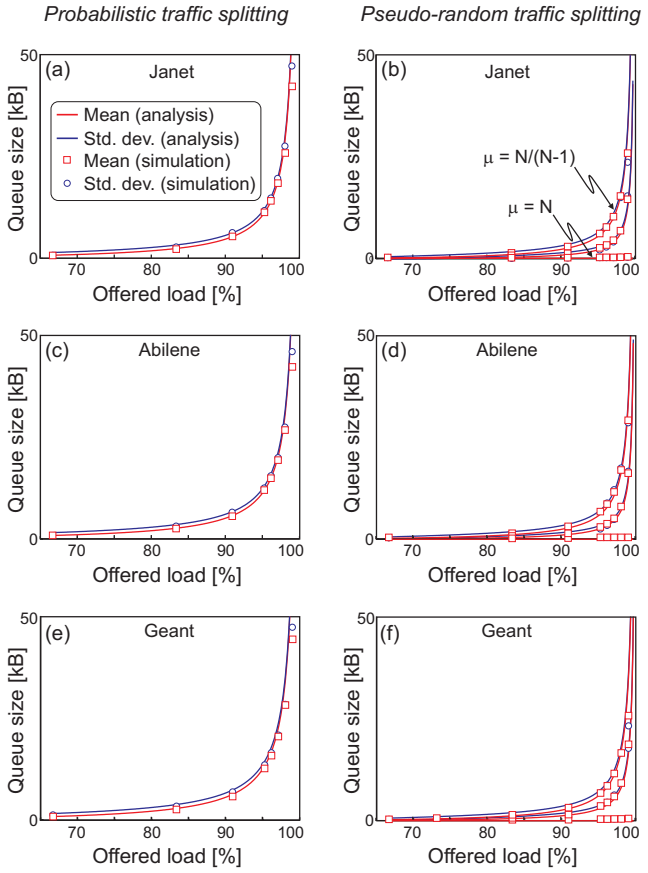


Fig. 7. Mean queue sizes and their standard deviations as a function of offered load for RLB using probabilistic step 1 traffic splitting (a,c,e) and pseudo-random splitting (b,d,f) on all three example networks. Symbols: simulation; lines: analyses.

behavior of RLB might be improved by implementing a priority mechanism between step 1 traffic and step 2 traffic when entering their common queue with service rate $2D/N$. To this end, we performed *ns2*-simulations for RLB with both probabilistic and pseudo-random step 1 traffic splitting where we assigned different priorities to the two traffic streams at each queue, favoring either step 1 traffic or step 2 traffic by different amounts. However, by doing so we did not observe any changes in the average queue size or its standard deviation. This **indifference to prioritization** is attributed to the lack of correlations in the packet arrival process.

C. Queue sizes versus offered load

Figure 7 shows the average queue sizes as well as their standard deviations as a function of offered load $(1-\varepsilon)$ for RLB using probabilistic step 1 traffic splitting (left column) and pseudo-random traffic splitting (right column) for all three of our example networks. The symbols represent *ns2*-simulations (squares: mean queue sizes; circles: standard deviations), and the lines are the analytic solutions obtained by our queuing analysis of Sec. III, Eqns. (10,11,12). **The theory is seen to be in excellent agreement with the simulations in all cases**.

As discussed in Secs. III and V-A, the traffic pattern has no impact on queue statistics for probabilistic step 1 traffic split-

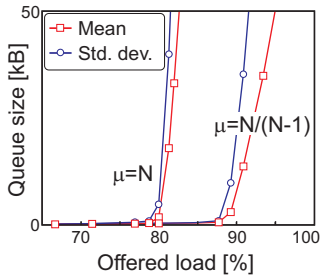


Fig. 8. Mean queue size and its standard deviation for SP routing on the Janet network assuming two different classes of hose traffic matrices.

ting, while it has for pseudo-random traffic splitting. Therefore, we plot three families of curves representing different matrix metrics for pseudo-random splitting: $\mu = N/(N-1)$, corresponding to uniform traffic matrices, $\mu = N$, corresponding to permutations of the identity matrix, as well as one value of μ falling midway in between. In agreement with our discussion in Sec. V-A, the queues remain empty for highly skewed traffic patterns ($\mu = N$), and for uniform traffic, the queuing behavior is close to the one for probabilistic splitting with half the slack parameter ε.

D. Comparison to shortest-path routing

Finally, Fig. 8 shows the performance of the shortest-path architecture on the Janet network for comparison. In order to set the link capacities for the SP network for a fair comparison with RLB, we first determined the link capacities that would be required to support all hose matrices on the network using the LP formalism described in Ref. 19, and then scaled back the link capacities such that the sum of all link capacities equaled the total capacity required for RLB. According to Tab. I, the scaling factor is 81%. We assumed traffic matrices that were close to uniform [$\mu = N/(N-1)$] as well as highly skewed ones [$\mu = N$]. As expected from our capacity scaling, queues start to build up at 81% load for SP routing under highly skewed traffic patterns. We observe a **strong dependence of queue build-up on the traffic matrix**, as well as a large standard deviation of the queue sizes across the network, indicating **severe hot spots in the network**. For larger networks, we observed even more pronounced differences as well as the expected instabilities resulting from a lack of capacity to support all demand patterns.

VI. Conclusions

We have studied the queueing behavior of randomized load balancing (RLB) across networks, using both analytical techniques as well as packet-level simulations based on *ns2*. Our results show that (*i*) for probabilistic traffic splitting, queueing delays are independent of the traffic pattern, (*ii*) for pseudo-random splitting, queueing delays are lower than for probabilistic splitting. For the latter case, queuing delays are provably worst for uniform traffic matrices and best for highly skewed matrices, which are becoming particularly important for emerging network applications. We have also shown that queuing behavior of RLB is uniform across the network, which in contrast to shortest-path routing avoids hot spots due to dynamically changing demand patterns.

Acknowledgment

We would like to acknowledge valuable discussions with F. Bruce Shepherd.

References

[1] W. Ben-Ameur and H. Kerivin, “New economical virtual private networks,” Commun. of the ACM 44(6), 69-73 (2003).
[2] J. Cao, K. Ramanan (2002). A Poisson limit for buffer overflow probabilities. In: *Proc. IEEE Infocom*.
[3] C.-S. Chang, D.-S. Lee, and Y.-S. Jou, “Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering,” 2001 IEEE Workshop on High Performance Switching and Routing (HPSR) (2001).
[4] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe, ”Resource management with hoses: Point-to-cloud services for virtual private networks,” IEEE/ACM Trans. on Networking 10(5), 679-692 (2002).
[5] J. Andrew Fingerhut, Subhash Suri and Jonathan Turner, “Designing Least-Cost Nonblocking Broadband Networks,” *Journal of Algorithms*, 287-309 (1997).
[6] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, ”Provisioning a virtual private network: a network design problem for multicommodity flow,” ACM Symposium on Theory of Computing (STOC’01) (2001).
[7] I. Keslassy, S.-T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, ”Scaling Internet Routers Using Optics,” ACM SIGCOMM 2003 (2003).
[8] J.F.C. Kingman (1970). Inequalities in the theory of queues. *J. Royal Stat. Soc. Series B* **32**, 102–110.
[9] L. Kleinrock (1976). *Queueing Systems Vol. II: Computer Applications*, Wiley & Sons, New York.
[10] M. Kodialam, T.V. Lakshman, and S Sengupta, ”Efficient and Robust Routing of Highly Variable Traffic,” HotNets III (2004).
[11] M. Kodialam, T. V. Lakshman and S. Sengupta, *Maximum Throughput Routing of Traffic in the Hose Model*, IEEE INFOCOM 2006.
[12] M. Kodialam, T. V. Lakshman, J. B. Orlin and S. Sengupta, *A Versatile Scheme for Routing Highly Variable Traffic in Service Overlays and IP Backbones*, IEEE INFOCOM 2006
[13] C.P. Kruskal, M. Snir, A. Weiss (1988). The distribution of waiting times in clocked multistage interconnection networks. *IEEE Trans. Comput.* **37**, 1337–1352.
[14] T. Leighton and S. Rao, “An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms”, in *Proc. 29th IEEE Symp. on Foundations of Computer Science (FOCS’88)*, 1988, pp. 422–431.
[15] B. Lin and I. Keslassy, *The Concurrent Matching Switch Architecture*, IEEE INFOCOM 2006.
[16] A. W. Marshall, I. Olkin (1979). Inequalities: Theory of Majorization and Its Applications. Academic Press, New York.
[17] D. Mitra, R.A Cieslak, “Randomized parallel communications on an extension of the omega network”, *J. ACM*, Vol.32, No.4 1987, 802-824.
[18] H.Nagesh, V.Poosala, V.Kumar, P.J.Winzer, and M.Zirngibl, ”Load-balanced architecture for dynamic traffic”, Optical Fiber Communication Conf. (OFC’05), Anaheim (CA/USA) OME67 (2005).
[19] F. B. Shepherd and P. J. Winzer, “Selective randomized load balancing and mesh networks with changing demands,” J. Opt. Netw. 5, 320-339 (2006).
[20] D. Stoyan (1983). *Comparison Methods for Queues and Other Stochastic Models*. John Wiley & Sons, Chichester.
[21] L. G. Valiant, ”A scheme for fast parallel communication”, SIAM J. Comput. 11(2), 350-361 (1982).
[22] I. Widjaja, A.I. Elwalid, “Exploiting parallelism to boost data-path rate in high-speed IP/MPLS networking, *Workshop on High-Speed Networking 2002*.
[23] R. Zhang-Shen and N. McKeown, ”Designing a Predictable Internet Backbone Network”, HotNets III, San Diego, CA, (2004).
[24] R. Zhang-Shen, M. Kodialam and T. V. Lakshman, *Achieving Bounded Blocking in Crcuit Switched Networks*, IEEE INFOCOM 2006