Optimal Load-Balancing

Isaac Keslassy* Cheng-Shang Chang† Nick McKeown* Duan-Shin Lee †

* Stanford University
Stanford, CA 94305-9030, USA
{keslassy,nickm}@stanford.edu

† National Tsing Hua University
Hsinchu 300, Taiwan, R.O.C.
{cschang@ee,lds@cs}.nthu.edu.tw

***Abstract*— This paper is about load-balancing packets across multiple paths inside a switch, or across a network. It is motivated by the recent interest in load-balanced switches. Load-balanced switches provide an appealing alternative to crossbars with centralized schedulers. A load-balanced switch has no scheduler, is particularly amenable to optics, and – most relevant here – guarantees 100% throughput. A uniform mesh is used to load-balance packets uniformly across all 2-hop paths in the switch. In this paper we explore whether this particular method of load-balancing is optimal in the sense that it achieves the highest throughput for a given capacity of interconnect. The method we use allows the load-balanced switch to be compared with ring, torus and hypercube interconnects, too. We prove that for a given interconnect capacity, the load-balancing mesh has the maximum throughput. Perhaps surprisingly, we find that the best mesh is slightly non-uniform, or biased, and has a throughput of $N/(2N-1)$, where N is the number of nodes.**

I. Introduction

A. From Scheduling to Load-Balanced Routing

Current Internet core routers commonly implement combined input and output queueing (CIOQ) with a centralized scheduler. Numerous centralized scheduling algorithms have been proposed in the literature [1], [2], [3], [4]. Nevertheless, although these scheduling algorithms can theoretically provide a guaranteed throughput of 50% to 100% ([5], [6], [7]), they are becoming impractical as the line rates and number of ports grow, because of their complexity and/or the speedup of the buffer memory.

There has been recent interest in a new approach, which eliminates scheduling, using a *load-balanced switch* architecture [8], [9], [10], [11], [12], [13], [14]. As shown in [12], this architecture appears to be a practical way to scale Internet routers to very high capacities, and achieve throughput guarantees for all traffic patterns.

Figure 1 shows the load-balanced switch architecture based on two fully-interconnected meshes, with $N = 4$ linecards interconnected by N^2 links. It consists of a single stage of buffers sandwiched by two identical stages of switching, where each switch is built from a uniform mesh. Each linecard in the first stage is connected to each linecard in the center stage by a channel at rate R/N, where R is the line rate and N is the number of linecards. Likewise, each linecard in the center stage is connected to each linecard in the final stage by a channel at rate R/N. The buffer at each center stage is partitioned into N virtual output queues (VOQs). To understand its operation, consider a stream of packets from a given input to a given output. The first mesh sends packets in round-robin to all intermediate inputs, load-balancing traffic across them. Each packet is put into the VOQ in the intermediate input according to its eventual output. The second mesh services each VOQ at fixed rate R/N, regardless of its occupancy. Each packet is transferred across the second mesh to its output, from where it departs the system. Thus, the two meshes work identically, but perform two different functions: the first one load-balances packets across the center stages, sending $1/N$-th of the traffic to each intermediate input, and the second one switches packets to their correct destination by servicing each VOQ at fixed rate R/N.

Although Figure 1 appears to show $3N$ linecards (N for

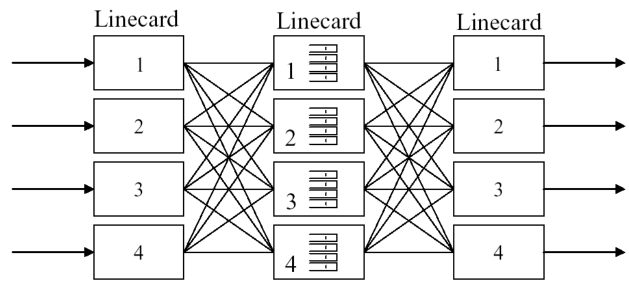

Fig. 1. Load-balanced switch architecture

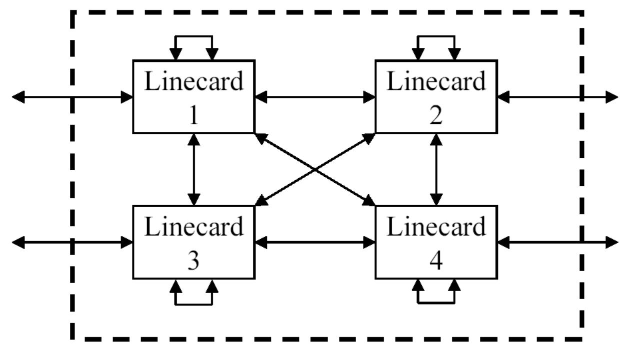

Fig. 2. Generic architecture of a load-balanced switch and of a load-balanced routing network

This research was supported by a Wakerly Stanford Graduate Fellowship, by the National Science Council, Taiwan, R.O.C., under contract NSC-91-2219-E007-003, by the Program for Promoting Academic Excellence of Universities, under contract 89-E-FA04-1-4, by the NSF Large ITR grant under contract NSF 02-168, by the DARPA/MARCO Center for Circuits, Systems and Software under MARCO contract 2001-CT-888 and DARPA grant MDA972-02-1-0004, and by Cisco Systems.

each stage), a real implementation would have N linecards, and each linecard would contain three logical parts (input, intermediate input and output). This means that the two meshes can be replaced by a single mesh running twice as fast, as shown in Figure 2. Every packet traverses the switch fabric twice: once from the input linecard to a VOQ in the intermediate linecard, then a second time from the VOQ to the output linecard.

B. The Throughput of Load-Balanced Switching

Perhaps the most interesting characteristic of the load-balanced switch is that it provably achieves 50% throughput (and therefore 100% throughput with a speedup of two) for a broad class of weakly mixing, stochastic arrivals [8]. Intuitively, the first stage makes traffic (just) uniform enough for the second stage to provide the throughput guarantee.

It is not immediately obvious why a load-balancing stage built from a uniform mesh with N inputs and outputs can make the traffic uniform enough, regardless of the traffic matrix or the burstiness of the arrivals. And it's even less obvious whether the mesh needs to be uniform (i.e. all links have the same capacity R/N); how does the throughput change if the mesh is non-uniform? What arrangement of link capacities maximizes the throughput?

More generally, we're interested in comparing the architecture with other well known ways to interconnect linecards. For example, a ring, a torus or a hypercube. We'll compare them by considering an interconnection network with a given total capacity. Packets are routed through the network to create a load-balanced switch, a ring, torus, or hypercube. We then determine which arrangement has the highest throughput.

To make the comparison, we'll use an arbitrary network with fixed capacities that we'll call a *load-balanced routing network*. As in the load-balanced switch, linecards are interconnected using a network with a fixed configuration and fixed capacities (Figure 2). Each incoming flow can be load-balanced across the different possible paths to its output, as long as the rate needed on each link is within its capacity. For each flow, a decision has to be made: how should it be load-balanced across the different possible paths?

Consider the example in Figure 3. It shows a simple load-balanced routing network where all the capacities between linecards are either zero (no link) or c. If linecard 1 wants to send traffic to linecard 4, it could send it directly using the link $1 \rightarrow 4$ (with capacity c). It could also choose to load-balance traffic using the paths $1 \rightarrow 2 \rightarrow 4$, $1 \rightarrow 3 \rightarrow 4$, or $1 \rightarrow 3 \rightarrow 2 \rightarrow 4$. We'll allow it to choose any path, even if it's obviously not useful, such as $1 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 4$ or $1 \rightarrow 1 \rightarrow 1 \rightarrow 4 \rightarrow 4$.

Essentially, what is normally a *scheduling* decision inside the router is transformed into a *routing* decision. While a centralized scheduler needs to decide how to configure a crossbar depending on the queue state, the linecards in a load-balanced routing network need to decide how to route flows across the different possible internal paths.

The general class of *load-balanced routing networks* appears in many areas of networking. Perhaps most commonly,

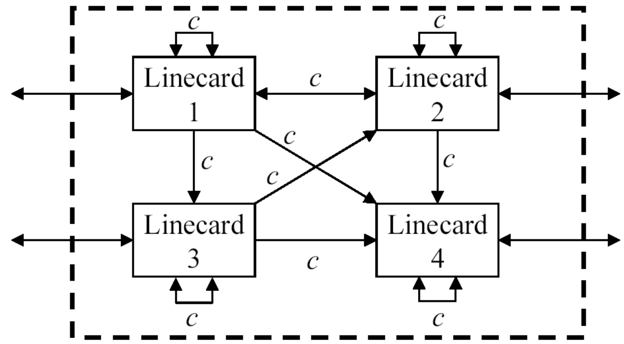

Fig. 3. Example of load-balanced routing network

load-balanced routing networks are an example of multipath routing [15] in a network, or Internet, of routers. They are also commonly used in torus and hypercube networks [16], [17] for the implementation of multi-stage, distributed switches inside routers [18], multiprocessor interconnection networks [19] and I/O interconnects [20]. For each flow, the path taken by the packets might then be pre-determined without regard to the state of the system (also called oblivious routing [21], which includes Valiant's randomized routing [22]); or adaptive (where routing is dependent on the queue state [23]). Load-balanced routing networks can also be used in fixed ad-hoc networks, such as sensor networks [24], [25]. Finally, load-balanced routing networks are a specific type of multi-commodity network that often appears in the networking literature. Understanding their theoretical bounds would be useful to the general class of multi-commodity network problems.

C. Main Results

We will analyze the throughput of load-balanced routing networks. The main findings of this paper are as follows. First, the throughput as a function of the capacity of load-balanced routing networks is concave, strictly increasing, and scales linearly. Second, a switch based on a uniform mesh has a guaranteed throughput of 50%, and so needs a speedup of two (or two meshes) to achieve 100% throughput. The uniform mesh is close, but not equal to, the interconnection with the highest throughput. A slightly biased, non-uniform mesh has a slightly higher throughput. In particular, the following is true:

Theorem 1: The capacity matrix with the best throughput exists and is unique. It is

$$\hat{C} = \frac{1}{2N-1} \cdot \begin{pmatrix} 1 & 2 & \dots & \dots & 2 \\ 2 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 2 \\ 2 & \dots & \dots & 2 & 1 \end{pmatrix},$$

and its throughput is $N/(2N-1) > 1/2$.

The reason is quite simple: In a uniform mesh, each node spreads traffic - and so routes packets - equally to all other nodes. But spreading to itself is redundant and inefficient. For instance, if node 1 has traffic to send to node 2 and the direct link $1 \rightarrow 2$ is congested, it can use load-balancing by sending part of this traffic to node 3, which will forward it to node 2. However, it is useless to send part of this traffic to node 1 for load-balancing, since this action just makes some packets come back to their starting point. Therefore, a link from a node to itself needs less capacity than a link from a node to another one, resulting in a non-uniform mesh. But asymptotically, for large N, the throughputs of the uniform and optimal meshes are the same.

In what follows we start by formulating more precisely the optimization problem in Section II, illustrate the definition of the guaranteed throughput in Section III, and provide its main properties in Section IV. Then, we describe the biased full mesh and compute its guaranteed throughput in Section V, show that its guaranteed throughput is optimal in Section VI, and prove that it is the only architecture with such a guaranteed throughput in Section VII. Finally, we analyze the load-balancing gain of an arbitrary architecture in Section VIII. All the proofs are in the Appendix.

II. Problem Formulation

A. Notations and Assumptions

Consider a network with N identical nodes, where $N \geq 2$. We define a *doubly stochastic* matrix to be a non-negative square matrix with all row and column sums equal to 1. Similarly, we define an *admissible* (or doubly sub-stochastic) matrix to be a non-negative square matrix with all row and column sums upper-bounded by 1. Finally, we define the time unit such that each node can send and receive at most one bit per second (if the maximum node speed is R, scale the time unit by a factor $\frac{1}{R}$).

A link of fixed capacity C_{ij} connects node i to node j, where $1 \leq i,j \leq N$. The matrix $C = [C_{ij}]_{1\leq i,j\leq N}$ is the capacity matrix, and any node l can send up to $\sum_{j=1}^{N} C_{lj}$ (and likewise receive at most $\sum_{i=1}^{N} C_{il}$) bits per time unit to and from the N nodes (including itself). Since every node l can send and receive at most one bit per time unit, $\sum_{i=1}^{N} C_{il} \leq 1$ and $\sum_{j=1}^{N} C_{lj} \leq 1$; therefore, the matrix C is admissible. The capacity matrix C defines the architecture; for example, the *uniform mesh architecture* (in which nodes are connected to each other with equal-capacity links), corresponds to the uniform matrix C where $C_{ij} = 1/N$. Similarly, a ring could be defined by $C_{ij} = \mathbf{1}_{\{j=i+1 \bmod N\}}$.

Denote by T the arrival traffic rate matrix, with T_{ij} being the arrival rate at node i of packets destined for node j. We will assume that T is admissible, since it cannot be supported otherwise. Suppose we want to load-balance these packets across multiple paths, each path having an arbitrary number of hops. If $P(i,j)$ is the set of paths between nodes i and j, then any path $p \in P(i,j)$ can be represented as $(i \rightarrow node_1 \rightarrow node_2 \rightarrow \ldots \rightarrow j)$. Let T_{ij}^p be the rate of the flow carried by p. If the arrival traffic rate matrix T is feasible (i.e., the network has 100% throughput for T), it is possible to decompose T into several paths p, and therefore for all i,j,

$$T_{ij} = \sum_{p \in P(i,j)} T_{ij}^p. \tag{1}$$

Similarly, we will define the effective load matrix L using for all i,j:

$$L_{ij} = \sum_{\{p:(i\rightarrow j)\in p\}} T_{ij}^p. \tag{2}$$

The effective load of a link is the sum of the loads of the paths sharing the link. A solution is feasible if and only if we can find a decomposition of T such that $L \leq C$, i.e., no link is over-booked.

B. Problem Intuition

Suppose that $N = 2$ and that we use a uniform mesh architecture, with capacity matrix

$$C = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}.$$

We will use this example to gain some intuition about the throughput of interconnection networks.

If the arrival rate matrix is

$$T_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0 \end{pmatrix}$$

then we cannot send traffic at rate 0.9 on the path $1 \rightarrow 1$, because the capacity is limited by $C_{11} = 0.5$. Therefore, *we need to load-balance the traffic by using the spare capacity of other links*. We will send 0.5 on the direct path $1 \rightarrow 1$, and the remaining 0.4 on the alternative path $1 \rightarrow 2 \rightarrow 1$. The resulting load matrix is

$$L_1 = \begin{pmatrix} 0.5 & 0.4 \\ 0.4 & 0 \end{pmatrix},$$

and $L_1 \leq C$. Clearly, the direct path is not always sufficient to carry the required rate matrix, but in this case it is possible to use a load-balanced path in order to carry it.

Not all rate matrices are feasible, i.e., the throughput is not always 100%. Consider the arrival rate matrix

$$T_2 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.9 \end{pmatrix}.$$

Sending 0.5 on $1 \rightarrow 1$, 0.4 on $1 \rightarrow 2 \rightarrow 1$, 0.5 on $2 \rightarrow 2$ and 0.4 on $2 \rightarrow 1 \rightarrow 2$, the load matrix is

$$L_2 = \begin{pmatrix} 0.5 & 0.8 \\ 0.8 & 0.5 \end{pmatrix},$$

and so $L_2 \not\leq C$. In this particular case, we need to scale down T_2 to

$$\begin{pmatrix} 0.75 & 0 \\ 0 & 0.75 \end{pmatrix}$$

for the solution to be feasible.

Finally, load-balancing does not always help, particularly in small matrices when there are not many paths to divert traffic away from congested links. And it is always useless to divert traffic to oneself. For example, consider the rate matrix

$$T_3 = \begin{pmatrix} 0 & 0.5+\epsilon \\ 0.5 & 0 \end{pmatrix},$$

where $\epsilon > 0$. Sending traffic on the path $1 \rightarrow 1 \rightarrow 2$ does not divert traffic from the congested link $1 \rightarrow 2$; therefore,

T_3 is not feasible. This teaches us that when sending traffic from node i to node $j \neq i$, it is clearly useless to use the link $i \rightarrow i$, because traffic is transferred across the network with no benefit. By comparing T_1, T_2 and T_3, this example also shows that finding the maximum throughput of a given rate matrix is not straightforward, even when $N = 2$. Moreover, since the number of cases to consider increases with N, such a problem is increasingly difficult to solve as N grows.

C. Problem Definition

Our objective is to find the load-balanced network with the largest throughput guarantee. In other words, we want to find a network with a guaranteed throughput θ^*, where θ^* satisfies two properties. First, given any admissible arrival traffic, the network guarantees a throughput θ^*, i.e., it will switch a fraction θ^* of the traffic for any input-output flow. And second, no other network can have a better guaranteed throughput than θ^*. We will define the problem by decomposing it into three successive optimization problems. First, we will find the throughput for a given network and a given rate matrix. Then, we will obtain the worst-case throughput of a network, which can be achieved for any rate matrix. Finally, we will provide θ^*, which is the best guaranteed throughput among all networks.

In the first optimization, we want to find the maximum throughput for a given network and a given rate matrix. In other words, given capacity matrix C and rate matrix T, we want to find the best possible throughput $\theta(C, T)$, such that the scaled-down rate demand matrix $\theta(C, T) \times T$ is feasible. Put mathematically,

$$\theta(C, T) \equiv \max_{\theta}(\theta), \text{ subject to:}$$

$$\begin{array}{lll} (i) & \sum_{p=1}^{P(i,j)} T_{ij}^p = \theta \times T_{ij} & \forall i, j \\ (ii) & L(i,j) \equiv \sum_{\{p:(i \rightarrow j) \in p\}} T_{ij}^p \leq C_{ij} & \forall i, j \\ (iii) & T_{ij}^p \geq 0 & \forall i, j, p \end{array}$$

In words, the throughput $\theta(C, T)$ is the maximum of the set of throughputs θ that satisfy three feasibility conditions. First, the arriving traffic is a scaled-down version of T by a factor θ, such that it can be decomposed into several paths p. The second condition is that the sum of the loads of the paths must be less than C, i.e., that the load matrix is feasible. The last condition is that the rate on each path must be nonnegative.

The second optimization finds the guaranteed maximum throughput $\theta(C)$ for the network. This is the throughput that is achievable by any rate matrix in the network, and, therefore,

$$\theta(C) \equiv \min_{T \text{ admissible}} (\theta(C, T)). \tag{3}$$

Finally, we find the maximum possible guaranteed throughput for any network, yielding a guaranteed throughput θ^*, where

$$\theta^* \equiv \max_{C \text{ admissible}} (\theta(C)). \tag{4}$$

III. Examples of Guaranteed Throughput

A. Guaranteed Throughput of the Uniform Mesh

The uniform mesh is an architecture in which all links have the same capacity, i.e., $C_{ij} = 1/N$ for all i, j. We will show that the maximum guaranteed throughput of the uniform mesh is 50%.

We saw already in the Introduction why the uniform mesh guarantees at least 50% throughput, although the proof was based on slightly different assumptions. In short, each packet goes through both the load-balancing stage and the forwarding stage, and therefore through two hops. Consequently, the link between node i and node j can receive load in two possible ways. Either node i is sending traffic to some node k and spreads it using the intermediate node j, or some node l sends traffic to node j and spreads it using the intermediate node i. Mathematically, $L_{ij} = \sum_k T_{ik} + \sum_l T_{lj} \leq 2$ with an admissible T. Therefore, $\theta(C) \geq 50\%$.

The following example shows that it is not possible to do better using a different load-balanced routing algorithm.

Assume that

$$T = \begin{pmatrix} 0 & x & 0 & \dots & 0 \\ 0 & 0 & x & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & & & 0 & x \\ x & 0 & \dots & 0 & 0 \end{pmatrix},$$

where $x \geq 1/2$. A node i can send at most $C_{i(i+1 \bmod N)} = 1/N$ amount of traffic directly.[1] It also needs to send the remaining $x - 1/N$ amount of traffic to load-balanced paths, with each of these paths using at least two links. Hence, the total traffic load contributed by each node to the system is at least $(1/N) + 2(x - 1/N)$, which implies that the total traffic load contributed by the N nodes is $N(1/N + 2(x - 1/N)) = 2Nx - 1$. As we saw earlier, diagonal elements do not help load-balancing, and with this rate matrix they are also useless for direct paths. Hence, the total useful traffic capacity is the sum of all non-diagonal elements of C, i.e., $N \cdot (1 - 1/N) = N - 1$. For the solution to be feasible, we need $2Nx - 1 \leq N - 1$, which translates into $x \leq 1/2$. And so there exists a traffic rate matrix that is only feasible with a throughput of at most 50%. This implies $\theta(C) \leq 50\%$. Since we found that the two-hop algorithm provides a throughput of 50%, it follows that

$$\theta(C) = 50\%. \tag{5}$$

Further, it is not possible to improve on the two-hop algorithm.

B. Guaranteed Throughput of a Ring

As a second example, consider a network in which the nodes are connected in a uni-directional ring, i.e., node i is connected to node $(i + 1) \bmod N$. Recall that we assumed that each packet needs to go at least once through the network. In the worst case, T is the identity matrix so that nodes only send

[1]The modulo function takes values in $\{1, ..., N\}$ when nodes are numbered $\{1, ..., N\}$.

traffic to themselves through the ring. Therefore, all packets cross N links, and the throughput $\theta(C_{ring}, T)$ is equal to $1/N$. This T is the worst case, since packets do not need to use more than N links to reach their destination. Therefore,

$$\theta(C_{ring}) = 1/N, \tag{6}$$

which — as expected — is much lower than for the uniform mesh.

C. Guaranteed Throughput of a Permutation Matrix

The ring is a special case of a permutation matrix σ of the set $\{1, ..., N\}$, where σ is the capacity matrix of a network. The matrix σ can be represented as a $0-1$ matrix with exactly one 1 in each row and column; i.e., $\sigma_{ij} = 1$ if $\sigma(i) = j$, and $\sigma_{ij} = 0$ otherwise. Since σ is a permutation, it can be decomposed as a product of disjoint cycles (the decomposition is unique up to the order of the cycles).

If σ can be written as a single cycle of length N, we can assume without loss of generality that $\sigma(1) = 2$, $\sigma(2) = 3$,..., $\sigma(N) = 1$, and so σ is the capacity matrix of a ring, with $\theta(\sigma) = 1/N$.

Alternatively, if σ can be written as the product of two or more cycles, then there are two nodes i and j such that node i is in the first cycle and node j is in the second one. It is then impossible to reach node j from node i (the capacity graph is not connected), hence the throughput for any matrix T such that $T_{ij} = 1$ is zero, and $\theta(\sigma) = 0$.

This example illustrates that the throughput of a capacity matrix is sensitive to its coefficients; and that the throughput of a disconnected graph is zero.

IV. Properties of the Guaranteed Throughput

In the above examples, we computed the throughputs of several capacity matrices, but found that it is not straightforward in general to compute throughput directly. Since we want to find the capacity matrix with the largest guaranteed throughput, we will use general properties of the throughput function. We will start by showing that it is concave in C, scales linearly, and is strictly increasing.

A. Concavity

First, we show that throughput is concave in C. Assume that two capacity matrices C_1 and C_2 achieve throughputs of $\theta(C_1, T)$ and $\theta(C_2, T)$ for a rate matrix T. Then, applying the definition of throughput, for any $\lambda \in [0, 1]$, the matrix $C = \lambda C_1 + (1-\lambda)C_2$ will achieve a throughput of $\theta(C, T) \geq \lambda\theta(C_1, T) + (1-\lambda)\theta(C_2, T)$. This can be seen by using the paths from C_1 for a fraction λ of the traffic, and the paths from C_2 for a fraction $1-\lambda$. As a consequence, we also have $\theta(C) \geq \lambda\theta(C_1) + (1-\lambda)\theta(C_2)$. This leads to the following proposition.

Proposition 2: The guaranteed throughput function $\theta(C)$ is concave in C.

B. Linear Scaling

Given any positive λ, we can find a feasible rate allocation for λC from the rate allocation for C (and vice versa) by scaling the rate assigned to each path by a factor λ (respectively by $\frac{1}{\lambda}$). Therefore, we get the following proposition:

Proposition 3: The guaranteed throughput function θ is linear with respect to scaling, i.e.,

$$\theta(\lambda \cdot C) = \lambda \cdot \theta(C).$$

C. Strictly Increasing

Clearly θ is a non-decreasing function in the space of admissible capacity matrices. In other words, having more capacity cannot decrease the throughput. If C and D are two admissible capacity matrices, where $C \leq D$ (i.e., for all i, j, $C_{ij} \leq D_{ij}$, defining a partial order relation), then from the definition of θ: $\theta(C) \leq \theta(D)$.

Now, if $D > C$, there exists ϵ such that

$$D \geq C + \epsilon C_{\text{uniform}},$$

where C_{uniform} is the capacity matrix of the uniform mesh. Hence

$$\begin{aligned}
\theta(D) &\overset{(a)}{\geq} \theta((1+\epsilon)(\frac{1}{1+\epsilon}C + \frac{\epsilon}{1+\epsilon}C_{\text{uniform}})) \\
&\overset{(b)}{=} (1+\epsilon) \times \theta(\frac{1}{1+\epsilon}C + \frac{\epsilon}{1+\epsilon}C_{\text{uniform}})) \\
&\overset{(c)}{\geq} (1+\epsilon)(\frac{1}{1+\epsilon}\theta(C) + \frac{\epsilon}{1+\epsilon}\theta(C_{\text{uniform}})) \\
&\overset{(d)}{=} (1+\epsilon)(\frac{1}{1+\epsilon}\theta(C) + \frac{\epsilon}{1+\epsilon}\frac{1}{2})) \\
&> \theta(C),
\end{aligned}$$

where (a) uses the fact that θ is non-decreasing, (b) uses the equality $\theta(\lambda \cdot C) = \lambda\theta(C)$, (c) uses the concavity of θ and (d) uses the value of $\theta(C_{\text{uniform}})$. Therefore, we obtain:

Proposition 4: The guaranteed throughput function θ is strictly increasing, i.e., if $C < D$ then $\theta(C) < \theta(D)$.

V. The Biased Mesh

A. Definition

We have already seen that the uniform mesh has a throughput of 50%, even though a node potentially spreads traffic over the useless links to itself. We can therefore expect a modified mesh — i.e., a mesh that does not spread traffic to itself — to have higher throughput. This is indeed the case; in fact, it is the network with the highest guaranteed throughput.

In this modified mesh, a link from a node to itself is only used to send traffic directly, and not for spreading. However, a link from a node to another one is used for sending traffic directly as well as for spreading. Therefore, intuitively, a link from a node to another one should have twice as much capacity as a link from a node to itself, because it will be used for two functions instead of one. We will call such a modified mesh the *biased mesh*. Its capacity matrix $\hat{C}$ is given by

$$\hat{C} = \begin{pmatrix} c & 2c & \dots & \dots & 2c \\ 2c & c & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & c & 2c \\ 2c & \dots & \dots & 2c & c \end{pmatrix},$$

where $c = 1/(2N-1)$.

In the remainder (Propositions 6, 7 and 8), we will show that $\hat{C}$ uniquely achieves the highest guaranteed throughput, using three consecutive steps. First, we will show that $\hat{C}$ achieves a throughput of $N/(2N-1)$. Then, we will prove that this is the largest achievable throughput for any network. Finally, we will demonstrate that the biased mesh is the only network to achieve this throughput.

B. Guaranteed Throughput of the Biased Mesh

Our first objective is to show that the guaranteed throughput of the biased mesh with the capacity matrix $\hat{C}$ is at least $N/(2N-1)$. Using the definition of the guaranteed throughput, we need to consider all admissible rate matrices T. The following proposition significantly restricts the number of rate matrices T we need to consider. It is proved in Appendix I.

Proposition 5: The guaranteed throughput $\theta(C)$ defined in (3) can be found by considering the set of permutation matrices, i.e.,

$$\theta(C) = \min_{T \text{ permutation}} (\theta(C,T)). \tag{7}$$

Proposition 5 restricts to the set of permutation matrices the set of rate matrices we need to consider. To show that the throughput of $\hat{C}$ is at least $N/(2N-1)$, we just need to show that a throughput of $N/(2N-1)$ can be achieved for all the permutation matrices. It leads to the following proposition, proved in Appendix I.

Proposition 6: The guaranteed throughput of the biased mesh with capacity matrix $\hat{C}$ is at least $N/(2N-1)$.

VI. Optimality of the Biased Mesh

We have just found that the biased mesh guarantees a throughput of at least $\frac{N}{2N-1}$. The following proposition shows that the biased mesh achieves the maximum possible guaranteed throughput for any admissible capacity matrix.

Proposition 7: If the capacity matrix C is admissible, then its guaranteed throughput satisfies

$$\theta(C) \leq \frac{N}{2N-1}.$$

The proof for Proposition 7 is in Appendix II.

VII. Uniqueness of the Optimal Capacity Matrix

Since we proved that the biased mesh achieves the optimal throughput $N/(2N-1)$, we will now demonstrate that it is the only capacity matrix to do so. This is done in Proposition 8, proved in Appendix III.

Proposition 8: The only capacity matrix C that can achieve the optimal throughput $N/(2N-1)$ is the capacity matrix $\hat{C}$ of the biased mesh.

In conjunction with Propositions 6, 7 and 8, we have, therefore, established the following theorem.

Theorem 9: The biased mesh satisfies the following three properties:

(i) The guaranteed throughput of the biased mesh is equal to $\hat{\theta} = N/(2N-1)$.

(ii) The biased mesh achieves the maximum possible guaranteed throughput for any network, i.e., $\theta(\hat{C}) = N/(2N-1)$.

(iii) The biased mesh is the only network to achieve this guaranteed throughput, i.e., $\theta(C') < \theta(\hat{C})$ for any admissible capacity matrix $C' \neq \hat{C}$.

VIII. The Benefit of Load-Balancing

We can now quantitatively analyze the benefits of load-balancing in an *arbitrary* network. Put mathematically, we can estimate the ratio of the guaranteed throughputs that can be achieved when load-balancing is allowed and when it is not. We will call this ratio the *load-balancing gain*.

A. Guaranteed Throughput without Load-Balancing

Let's compute the guaranteed throughput without load-balancing, when only direct links can be used. To go from node i to node j, a packet must be sent over the link between i and j, and cannot be load-balanced via a third node. In general, the guaranteed throughput of a non-load-balanced network will be determined by its weakest link, as can be seen when using a rate matrix that fully uses the weakest link. Thus, the guaranteed throughput of a capacity matrix C will be

$$\min_{i,j} C_{ij}.$$

For instance, without load-balancing, the guaranteed throughput of the uniform full mesh is $1/N$, and the guaranteed throughput of the biased full mesh is $1/(2N-1)$.

B. Guaranteed Throughput with Load-Balancing

Let's now bound the guaranteed throughput with load-balancing so as to bound the benefit of load-balancing. From Theorem 9, we know that the guaranteed throughput of a network is upper-bounded by $\hat{\theta}$, but we need to find a lower-bound on the guaranteed throughput. We can do it by comparing the network to the biased full mesh, which has the highest guaranteed throughput. Using the linear scaling and monotonicity properties of the throughput function, we find that for any $\lambda \in [0,1]$,

$$C \geq \lambda \hat{C} \Rightarrow \theta(C) \geq \lambda \hat{\theta}.$$

In other words, if a given network has at least as much capacity as the scaled-down version of the biased full mesh, then it will also provide at least as much guaranteed throughput as the scaled-down guaranteed throughput of the biased full mesh. We can obtain the following proposition:

Proposition 10: The guaranteed throughput $\theta(C)$ for any capacity matrix C satisfies:

$$\hat{\theta} \cdot \min_{i,j} \left(\frac{C_{ij}}{\hat{C}_{ij}} \right) \leq \theta(C) \leq \hat{\theta}.$$

C. Load-Balancing Gain

Define the load-balancing gain as the ratio of the guaranteed throughputs with and without load-balancing. Mathematically,

$$l.b.gain \triangleq \frac{\theta(C)}{\min_{i,j} C_{ij}}.$$

The load-balancing gain is a measure of the gain in throughput guarantee achieved by load-balancing. The following Proposition provides bounds on the load-balancing gain. It is proved in Appendix IV.

Proposition 11: The load-balancing gain for any capacity matrix C satisfies the following bounds:

$$\frac{N}{2} \leq l.b.gain \leq \frac{\hat{\theta}}{\min_{i,j} C_{ij}}.$$

Therefore, load-balancing always improves guaranteed throughput by a factor of at least $N/2$.

The upper-bound on the load-balancing gain reflects the fact that a system is forced to rely heavily on load-balancing when its weakest link cannot carry enough capacity.

For example, let's apply these bounds to the uniform full mesh and the biased full mesh. For the uniform full mesh, the lower-bound is tight, and Proposition 11 becomes:

$$\frac{N}{2} \leq l.b.gain \equiv \frac{N}{2} \leq \frac{N}{2} \cdot \frac{1}{1 - \frac{1}{2N}}.$$

For the biased full mesh, the upper-bound is tight, and Proposition 11 becomes:

$$\frac{N}{2} \leq l.b.gain \equiv N \leq N.$$

As an aside, it is interesting to note that since the uniform mesh achieves 50% throughput (Equation 5), we know that the uniform mesh is

$$\begin{aligned} \frac{\theta(\hat{C})}{\theta(C_{\text{uniform}})} &= \frac{\frac{N}{2N-1}}{\frac{1}{2}} = \frac{1}{1 - \frac{1}{2N}} \\ &= 1 + o(1) \text{ — optimal} \end{aligned}$$

for its guaranteed throughput. Therefore, the load-balanced switch with a uniform mesh is *asymptotically optimal*. Asymptotically with N, it guarantees at least as much throughput as any other fixed interconnection with an admissible capacity matrix.

IX. Conclusion

When building a router or network we can choose from among many different interconnection topologies; and can choose whether or not to use load-balancing. In different situations, we might want the network to have different properties; for example, minimize packet delay, maximize network scalability or ensure no single-point of failure. In this paper we assumed that we want to maximize the throughput in a system for which we don't know *a priori* what the traffic matrix will be.

We can summarize our main result by saying that while some networks are difficult to analyze (e.g. the throughput of sensor networks with arbitrary topology [24], [25]), and some packet routing algorithms are difficult to analyze (e.g. the throughput of adaptive algorithms [23]), we can say that when the traffic matrix is not known, the guaranteed throughput of a biased full mesh will *always be strictly better* than the guaranteed throughput of any other network using any routing algorithm.

This is quite a strong result, and should provide guidance to those designing router interconnects, network topologies, and multipath routing algorithms.

References

[1] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi and F. Neri, "Packet scheduling in input-queued cell-based switches," *Proc. of IEEE Infocom '01*, Anchorage, Alaska, April 2001.

[2] Y. Tamir and H.C. Chi, "Symmetric crossbar arbiters for VLSI communication switches," *IEEE Trans. on Parallel and Distributed Systems*, vol. 4, no. 1, pp. 13-27, 1993.

[3] N. McKeown, "iSLIP: A Scheduling Algorithm for Input-Queued Switches" *IEEE Transactions on Networking, Vol 7, No.2*, April 1999.

[4] T.E. Anderson, S.S. Owicki, J.B. Saxe, and C.P. Thacker, "High speed switch scheduling for local area networks," *ACM Trans. on Computer Systems*, Vol. 11, No. 4, pp. 319-352, Nov. 1993.

[5] N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. on Comm.*, Vol. 47, No. 8, Aug. 1999.

[6] J.G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," *Proc. of the IEEE INFOCOM*, Vol. 2, pp. 556-564, Tel Aviv, Israel, March 2000.

[7] E. Leonardi, M. Mellia, F. Neri and M. A. Marsan, "On the stability of input-queued switches with speed-up," *IEEE/ ACM Transactions on Networking*, Vol. 9, No. 1, pp. 104-118, Feb. 2001.

[8] C.S. Chang, D.S. Lee and Y.S. Jou, "Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering," *IEEE HPSR '01*, Dallas, May 2001.

[9] C.S. Chang, D.S. Lee and C.M. Lien, "Load balanced Birkhoff-von Neumann switches, Part II: multi-stage buffering," *Computer Comm.*, Vol. 25, pp. 623-634, 2002.

[10] I. Keslassy and N. McKeown, "Maintaining packet order in two-stage switches," *Proc. of the IEEE Infocom*, June 2002.

[11] C.S. Chang, D.S. Lee and C.Y. Yue, "Providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches," *Proc. of the IEEE Infocom*, 2003.

[12] I. Keslassy, S.-T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard and N. McKeown , "Scaling Internet routers using optics," *ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[13] C.S. Chang, D.S. Lee and Y.J. Shih , "Mailbox switch: a scalable two-stage switch architecture for conflict resolution of ordered packets," *Proceedings of IEEE Infocom '04*, Hong Kong, March 2004.

[14] I. Keslassy, S.T. Chuang, N. McKeown, "A Load-Balanced Switch with an Arbitrary Number of Linecards," *Proceedings of IEEE Infocom '04*, Hong Kong, March 2004.

[15] S. Vutukury, "Multipath routing mechanisms for traffic engineering and quality of service in the Internet," *PhD Thesis*, March 2001.

[16] William J. Dally, "Performance analysis of k-ary n-cube interconnection networks," *IEEE Transactions on Computers*, Vol. C-39, No. 6, pp. 775-785, June 1990.

[17] M. D. Grammatikakis, D. F. Hsu, M. Kraetzl, and J. Sibeyn, "Packet routing in fixed-connection networks: a survey," *Journal of Parallel and Distributed Processing*, Vol. 54(2), pp. 77-132, 1998.

[18] W. Dally, P. Carvey, and L. Dennison, "Architecture of the Avici terabit switch/router," *Proc. Hot Interconnects VI*, pp. 4150, Aug. 1998.

[19] S. Scott and G. Thorson, "The Cray T3E network: adaptive routing in a high performance 3D torus," *Proc. Hot Interconnects IV*, Aug. 1996.

[20] G. Pfister, "An Introduction to the InfiniBand Architecture," *High Performance Mass Storage and Parallel I/O*, IEEE Press, 2001.

[21] B. Towles and W.J. Dally, "Worst-case traffic for oblivious routing functions," *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, Winnipeg, Manitoba, Canada, Aug. 2002.

[22] L. Valiant and G. Brebner, "Universal schemes for parallel communication," *Proc. of the 13th annual symposium on theory of computing*, pp. 263-277, May 1981.

[23] A. Singh, W.J. Dally, A.K. Gupta and B. Towles, "GOAL: A Load-Balanced Adaptive Routing Algorithm for Torus Networks," *International Symposium on Computer Architecture (ISCA)*, San Diego, CA, USA, June 2003.

[24] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," *Proceedings of Mobile Computing and Networking (MOBICOM '99)*, Washington, Aug. 1999.

[25] S. Tilak, N. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless microsensor network models," *ACM Mobile Computing and Communications Review (MC2R)*, 2002.

[26] C.S. Chang, J.W. Chen and H.Y. Huang, "On service guarantees for input-buffered crossbar switches: a capacity decomposition approach by Birkhoff and Von Neumann," *IEEE IWQoS, London*, 1999.

[27] J. von Neumann, "A certain zero-sum two-person game equivalent to the optimal assignment problem," *Contributions to the Theory of Games*, vol. 2, pp. 5-12, Princeton University Press, Princeton, NJ, 1953.

[28] G. D. Birkhoff, "Tres observaciones sobre el algebra lineal," *Universidad Nacional de Tucuman Revista*, Serie A, vol. 5, pp. 147-151, 1946.

APPENDIX I
GUARANTEED THROUGHPUT OF THE BIASED MESH

A. Proof of Proposition 5

Proof: For any admissible matrix T, there is at least one doubly stochastic matrix $\overline{T}$ such that $T \leq \overline{T}$ [26], [27]. Clearly $\theta(C, \overline{T}) \leq \theta(C, T)$, and so we only need to consider the doubly stochastic rate matrices.

Birkhoff's theorem states that the set of doubly stochastic matrices equals the convex hull of the permutation matrices [28]. The claimed result follows from the definition of throughput. ∎

B. Proof of Proposition 6

Proof: We will prove that $\hat{C}$ achieves a throughput of $N/(2N-1)$ when $T = \sigma$, with σ a permutation. Let $c = 1/(2N-1)$. We consider a node i, and prove that i can always send at rate Nc to $\sigma(i)$. Our objective is to send as much flow as we can directly, and to uniformly load-balance the remainder among the non-diagonal elements. We distinguish two cases: either $\sigma(i) = i$ or $\sigma(i) \neq i$.

If $\sigma(i) = i$, node i needs to send Nc to itself. Therefore, node i can send c directly to itself, and load-balance the remaining rate of $(N-1)c$ among the other $(N-1)$ nodes, then sending c to each node.

If $\sigma(i) \neq i$, node i needs to send Nc to node $\sigma(i) \neq i$. Therefore, node i can send $2c$ directly to $\sigma(i)$, and load-balance the remaining rate of $(N-2)c$ among the $(N-2)$ nodes different from i and $\sigma(i)$; and each such node then sends c again to node $\sigma(i)$.

Let us examine the load on each link. Each diagonal element $\hat{C}_{ii}$ only receives traffic if it is destined from node i to node i, and in this case it receives exactly c, its capacity.

Moreover, each non-diagonal element $\hat{C}_{ij}$ can only receive traffic in two distinct cases, which cannot happen at the same time. If $j = \sigma(i)$, $\hat{C}_{ij}$ receives exactly $2c$, its capacity. Otherwise $j \neq \sigma(i)$, and $\hat{C}_{ij}$ receives c from the load-balanced path $i \rightarrow j \rightarrow \sigma(i)$, and c from the load-balanced path $\sigma^{-1}(j) \rightarrow i \rightarrow j$, summing to $2c$, its capacity.

The load on each link is therefore always bounded by its capacity; hence, this solution is feasible and the guaranteed throughput of $\hat{C}$ is at least $Nc = N/(2N-1)$. ∎

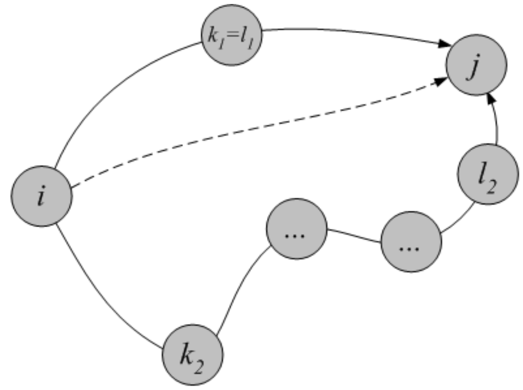

Fig. 4. Load-balancing example illustrating Proposition 12. The dashed line between i and j is a direct path. The two other paths are load-balanced paths in $P_{LB}(i,j)$. In the first load-balanced path $k_1 = l_1$, in the second $k_2 \neq l_2$.

APPENDIX II
OPTIMALITY OF THE BIASED MESH (PROPOSITION 7)

In this Appendix, we will prove that the biased mesh achieves the maximum possible guaranteed throughput for any possible admissible capacity matrix by establishing Proposition 7. To do so, we will first prove several useful Lemmas and Propositions by considering rate matrices that are also permutation matrices.

A. Throughput Bounds over the Set of Permutation Rate Matrices

It helps to study how the load-balancing is done. Let the set of *load-balanced paths*

$$P_{LB}(i,j) = \{p \in P(i,j) : (i \rightarrow j) \notin p\}$$

be the set of paths p between nodes i and j such that the link $i \rightarrow j$ is not in p. We will call paths not in $P_{LB}(i,j)$ *direct paths*. For instance, $1 \rightarrow 3 \rightarrow 2$ is a load-balanced path between node 1 and node 2, whereas $1 \rightarrow 2$ and $1 \rightarrow 1 \rightarrow 2 \rightarrow 2$ are direct paths.

Proposition 12: Any path $p \in P(i,j)$ satisfies one of the following two cases:

(i) If p is a direct path then $(i \rightarrow j) \in p$, or

(ii) If p is a load-balanced path then there exist two nodes k and l, possibly equal, such that $k \neq i$, $k \neq j$, $l \neq i$ and $l \neq j$, such that p contains $i \rightarrow k$ and $l \rightarrow j$.

Proof: (i) clearly follows from the definition of $P_{LB}(i,j)$. In (ii), by definition of $P_{LB}(i,j)$, at least one node is different from i, and if node k is the first node in path p that is different from i, then $k \neq j$ also. Similarly, if node l is the last node in path p that is different from j, then $l \neq i$ also. ∎

Using this characterization of load-balanced paths, we consider all the rate matrices that are also permutation matrices, such that each node sends all its traffic to some other node. For a permutation σ, let

$$S_1(\sigma) = \{i : \sigma(i) = i\}$$

denote the set of nodes invariant to σ, and let

$$S_2(\sigma) = \{i : \sigma(i) \neq i\} = \{1, ..., N\} \setminus S_1$$

denote the remaining nodes. The following lemma proves a general upper bound on the throughput $\theta(C)$ by considering the set of rate matrices that are permutation matrices.

Lemma 13: Given a capacity matrix C, the throughput $\theta(C)$ has the following upper bound taken over the set of permutations:

$$\theta(C) \leq \frac{1}{2} + \frac{1}{2N} \times \min_{\sigma \ permut.} \left(\sum_{i \in S_1(\sigma)} C_{ii} + \sum_{i \in S_2(\sigma)} (C_{i\sigma(i)} - C_{ii}) \right)$$

Proof: By definition of the throughput $\theta(C)$, for any permutation σ,

$$\theta(C) \leq \theta(C, \sigma).$$

Therefore, we only need to show that for any permutation σ,

$$\theta(C, \sigma) \leq \frac{1}{2} + \frac{1}{2N} \left(\sum_{i \in S_1(\sigma)} C_{ii} + \sum_{i \in S_2(\sigma)} (C_{i\sigma(i)} - C_{ii}) \right).$$

Consider a given permutation σ. By definition of $\theta(C, \sigma)$, any node i manages to send traffic at rate $\theta(C, \sigma)$ to node $j = \sigma(i)$. (We know that the optimum can be reached because the throughput θ is defined using continuous functions on compact sets.)

Consider then a path p between i and $j = \sigma(i)$, and distinguish between the following cases.

1) If $p \notin P_{LB}(i, j)$, then from Proposition 12, p contributes at least T_{ij}^p to C_{ij}.
2) If $p \in P_{LB}(i, j)$, then by Proposition 12 there exists two nodes k and l such that $k \neq i$, $k \neq j$, $l \neq i$ and $l \neq j$, and such that p contains $i \to k$ and $l \to j$. Hence, p will use a rate of at least T_{ij}^p out of the capacity C_{ik} in order to carry the link $i \to k$, and will also use a rate of at least T_{ij}^p out of the capacity C_{lj} in order to carry the link $l \to j$.
   Therefore, p requires a total rate of at least $2 \cdot T_{ij}^p$ from the non-diagonal elements of the capacity matrix C.

These two cases show that the link between i and $j = \sigma(i)$ can use non-diagonal capacity both with direct and load-balanced paths.

In particular, the first case studies the *direct* paths. It shows that the link between i and $j = \sigma(i)$ uses a rate of at least $\sum_{p \notin P_{LB}(i,\sigma(i))} T_{i\sigma(i)}^p$ out of $C_{i\sigma(i)}$ for the direct paths. This is a contribution to the non-diagonal capacity if and only if $\sigma(i) \neq i$, i.e., $i \in S_2(\sigma)$. Also, since the capacity for the direct link should be greater than its rate in order to be feasible, we get

$$C_{i\sigma(i)} \geq \sum_{p \notin P_{LB}(i,\sigma(i))} T_{i\sigma(i)}^p. \tag{8}$$

The second case studies the *load-balanced* paths. It shows that the link between i and $j = \sigma(i)$ uses a rate of at least $\sum_{p \in P_{LB}(i,\sigma(i))} 2 \cdot T_{i\sigma(i)}^p$ out of the non-diagonal elements of C for the load-balanced paths.

As a feasibility condition, the sum of the capacities of all the non-diagonal links should be more than the sum of all the rates required from these non-diagonal links. Therefore, using the two cases studied above, we get

$$\text{non-diagonal capacity} \geq \text{non-diagonal required rate},$$

i.e.,

$$\sum_{i,j \neq i} C_{ij} \geq \sum_{i \in S_2(\sigma)} \sum_{p \notin P_{LB}(i,\sigma(i))} T_{i\sigma(i)}^p + \sum_{i=1}^{N} \sum_{p \in P_{LB}(i,\sigma(i))} 2T_{i\sigma(i)}^p.$$

We now study the two sides of this equation. On the left hand side, since C is admissible, we have

$$N - \sum_i C_{ii} \geq \sum_{i,j \neq i} C_{ij}.$$

On the right hand side, the sum of all the rates required from these non-diagonal links can be rewritten as

$$\begin{aligned} & \sum_{i \in S_2(\sigma)} \sum_{p \notin P_{LB}(i,\sigma(i))} T_{i\sigma(i)}^p \\ & + \left[\sum_{i=1}^{N} \left(\sum_{p \notin P_{LB}(i,\sigma(i))} 2T_{i\sigma(i)}^p + \sum_{p \in P_{LB}(i,\sigma(i))} 2T_{i\sigma(i)}^p \right) - \sum_{i=1}^{N} \sum_{p \notin P_{LB}(i,\sigma(i))} 2T_{i\sigma(i)}^p \right] \\ = \; & 2N\theta(C,\sigma) - 2 \sum_{i \in S_1(\sigma)} \sum_{p \notin P_{LB}(i,\sigma(i))} T_{i\sigma(i)}^p \\ & - \sum_{i \in S_2(\sigma)} \sum_{p \notin P_{LB}(i,\sigma(i))} T_{i\sigma(i)}^p, \end{aligned}$$

using $T = \theta(C, \sigma) \cdot \sigma$ and $S_1(\sigma) \cup S_2(\sigma) = \{1, ..., N\}$ in the last equality. Using Equation (8), the sum of the non-diagonal rates can therefore be lower bounded by

$$2N\theta(C, \sigma) - 2 \sum_{i \in S_1(\sigma)} C_{i\sigma(i)} - \sum_{i \in S_2(\sigma)} C_{i\sigma(i)}.$$

Finally, we combine the equations and use the definition of $S_1(\sigma)$: $i \in S_1(\sigma)$ iff $\sigma(i) = i$. We get

$$N - \sum_i C_{ii} \geq 2N\theta(C, \sigma) - 2 \sum_{i \in S_1(\sigma)} C_{ii} - \sum_{i \in S_2(\sigma)} C_{i\sigma(i)}.$$

Therefore,

$$\theta(C, \sigma) \leq \frac{1}{2} + \frac{1}{2N} \left(\sum_{i \in S_1(\sigma)} C_{ii} + \sum_{i \in S_2(\sigma)} (C_{i\sigma(i)} - C_{ii}) \right).$$

■

B. Throughput Upper-Bound for a Capacity Matrix

We will now provide an upper-bound for the throughput of a capacity matrix by considering specific permutations. For $0 \leq k \leq N - 1$, define the permutation σ_k as the k^{th} sub-diagonal, i.e., assume that node i destines all its traffic to $\sigma_k(i) = i+k \bmod N$. We can then apply Lemma 13 to find the upper bound corresponding to each permutation, as expressed in the following lemma.

Lemma 14: Given a capacity matrix C, the throughput $\theta(C)$ has the following upper bounds:

$$\theta(C) \leq \frac{1}{2} + \frac{\sum_{i=1}^{N} C_{ii}}{2N}, \quad (9)$$

and

$$\theta(C) \leq \frac{1}{2} + \frac{\min_{1\leq k\leq N}(\sum_{i=1}^{N}(C_{i(i+k \bmod N)} - C_{ii}))}{2N}. \quad (10)$$

Proof: For $k = 0$, $S_1(\sigma_k) = \{1, ..., N\}$, hence the upper-bound from Lemma 13 is $\frac{1}{2} + \frac{\sum_{i=1}^{N} C_{ii}}{2N}$. Similarly, for $1 \leq k \leq N-1$, $S_2(\sigma_k) = \{1, ..., N\}$, hence this upper-bound is $\frac{1}{2} + \frac{\sum_{i=1}^{N}(C_{i\sigma_k(i)} - C_{ii})}{2N}$. ■

Proposition 15: If the capacity matrix C is admissible, i.e., C is a doubly sub-stochastic matrix, then its throughput satisfies

$$\theta(C) \leq \frac{N}{2N-1}.$$

Proof: We will prove this by contradiction. Suppose that $\theta(C) > \frac{N}{2N-1}$. For $0 \leq k \leq N-1$, let

$$x_k = \sum_{i=1}^{N} C_{i(i+k \bmod N)}.$$

It follows from (9) and (10) that

$$x_0 > \frac{N}{2N-1}, \quad (11)$$

and for $k = 1, 2, \ldots, N-1$,

$$x_k - x_0 > \frac{N}{2N-1}. \quad (12)$$

Therefore, we have $x_k > \frac{2N}{2N-1}$ for $k = 1, 2, \ldots, N-1$. Summing up for all k yields

$$N < \sum_{k=0}^{N-1} x_k = \sum_{i=1}^{N}\sum_{j=1}^{N} C_{i,j}.$$

This contradicts the assumption that the capacity matrix C is a doubly sub-stochastic matrix. ■

As the biased mesh with capacity matrix $\hat{C}$ achieves the throughput $N/(2N-1)$, it then follows from Proposition 15 that the biased mesh is optimal among all the admissible capacity matrices.

APPENDIX III
UNIQUENESS OF THE OPTIMAL CAPACITY MATRIX (PROPOSITION 8)

In this Appendix, we will prove that the biased mesh is the only capacity matrix that achieve the optimal throughput $N/(2N-1)$, and therefore we will be able to establish Proposition 8.

Lemma 16: If an admissible capacity matrix C achieves the optimal throughput $N/(2N-1)$, then the capacity matrix C satisfies

$$\sum_{i=1}^{N} C_{ii} = \frac{N}{2N-1}, \quad (13)$$

and for $k = 1, 2, \ldots, N-1$,

$$\sum_{i=1}^{N} C_{i(i+k \bmod N)} = \frac{2N}{2N-1}. \quad (14)$$

Proof: As in the proof of Proposition 7, let

$$x_k = \sum_{i=1}^{N} C_{i(i+k \bmod N)}.$$

If an admissible capacity matrix C achieves the optimal throughput $N/(2N-1)$, then we have from (9) and (10) that

$$x_0 \geq \frac{N}{2N-1}, \quad (15)$$

and for $k = 1, 2, \ldots, N-1$,

$$x_k \geq \frac{2N}{2N-1}. \quad (16)$$

If one of the inequalities in (15) and (16) is strict, then $\sum_{k=0}^{N-1} x_k$ will be strictly larger than N and this will contradict to the assumption that C is admissible. Therefore, we conclude that all the inequalities in (15) and (16) are in fact equalities. ■

Lemma 17: If an admissible capacity matrix C achieves the optimal throughput $N/(2N-1)$, then for any permutation σ,

$$\sum_{i\in S_2(\sigma)} (C_{i\sigma(i)} - 2C_{ii}) = 0.$$

Proof: Equation (13) in Lemma 16 provides $\sum_{i\in S_1(\sigma)} C_{ii} + \sum_{i\in S_2(\sigma)} C_{ii} = N/(2N-1)$ for any permutation σ. Hence, using Lemma 13, we get

$$\frac{N}{2N-1} \leq \frac{1}{2} + \frac{1}{2N}\min_{\sigma}\left(\frac{N}{2N-1} + \sum_{i\in S_2(\sigma)}(C_{i\sigma(i)} - 2C_{ii})\right),$$

where the minimum is taken over the set of permutation matrices. Therefore,

$$0 \leq \min_{\sigma}\left(\sum_{i\in S_2(\sigma)}(C_{i\sigma(i)} - 2C_{ii})\right),$$

i.e., for any permutation σ,

$$0 \leq \sum_{i\in S_2(\sigma)}(C_{i\sigma(i)} - 2C_{ii}).$$

We now use the fact that there are exactly $(N-1)!$ permutations σ such that $\sigma(i) = j$ for any nodes i and j. As a consequence, given a node i, there are exactly $(N-1)!$ permutations σ such that $i \notin S_2(\sigma)$, i.e., such that $\sigma(i) = i$. Therefore, there are exactly $N! - (N-1)!$ permutations σ

such that $i \in S_2(\sigma)$. We can deduce that

$$\begin{aligned}
&\sum_\sigma \left(\sum_{i \in S_2(\sigma)} (C_{i\sigma(i)} - 2C_{ii}) \right) \\
&= (N-1)! \sum_{i,j \neq i} C_{ij} - 2(N! - (N-1)!) \sum_i C_{ii} \\
&= (N-1)! \sum_{i,j} C_{ij} - (2N! - (N-1)!) \sum_i C_{ii} \\
&= N! - (N-1)! \cdot (2N-1) \cdot \frac{N}{2N-1} = 0,
\end{aligned}$$

where we use (13) in the last equality. Therefore, given that the sum of all these numbers is 0, and that they were all shown to be nonnegative, this means that they are all null. ∎

The next lemma enables us to determine the exact value of the diagonal elements of C.

Lemma 18: If an admissible capacity matrix C achieves the optimal throughput $N/(2N-1)$, then for all i,

$$C_{ii} = \frac{1}{2N-1}.$$

Proof: Pick arbitrarily any node — for instance, node 1 without loss of generality. For any node $j \neq 1$, consider the permutation σ such that $\sigma(1) = j$, $\sigma(j) = 1$, and the restriction of σ to the other elements is the identity. By Lemma 17, $C_{1j} + C_{j1} = 2(C_{11} + C_{jj})$. Summing over all such j's yields $\sum_{j=2}^N (C_{1j} + C_{j1}) = \sum_{j=2}^N 2(C_{11} + C_{jj})$. Adding $2C_{11}$ on each side of the equation and using (13) and (14) yields

$$1 + 1 = 2(N-1)C_{11} + \frac{2N}{2N-1}.$$

Hence $C_{11} = \frac{1}{2N-1}$. Since we picked the first node arbitrarily, this is similarly true for any node. ∎

Proposition 19: The only matrix C that can achieve the optimal throughput $N/(2N-1)$ is the capacity matrix $\hat{C}$ from the biased mesh.

Proof: Combining Lemmas 17 and 18, for any permutation σ, $\sum_{i \in S_2(\sigma)} C_{i\sigma(i)} = (2 \cdot |S_2(\sigma)|)/(2N-1)$, where $|S_2(\sigma)|$ denotes the number of elements in $S_2(\sigma)$.

Define matrix D such that $D_{ij} = C_{ij}$ for $i \neq j$, and $D_{ii} = 2/(2N-1) = 2C_{ii}$. Then all row and column sums of D are equal to $1 + 1/(2N-1)$ (because C is doubly stochastic). In addition, for any permutation σ,

$$\begin{aligned}
\sum_i D_{i\sigma(i)} &= \sum_{i \in S_1(\sigma)} D_{i\sigma(i)} + \sum_{i \in S_2(\sigma)} D_{i\sigma(i)} \\
&= \sum_{i \in S_1(\sigma)} D_{ii} + \sum_{i \in S_2(\sigma)} D_{i\sigma(i)} \\
&= \frac{2 \cdot |S_1(\sigma)|}{2N-1} + \frac{2 \cdot |S_2(\sigma)|}{2N-1} \\
&= \frac{2N}{2N-1}.
\end{aligned}$$

Hence, any permutation on D has the same sum! For any two nodes i, j, construct two permutations equal everywhere except on $\{D_{11}, D_{i1}, D_{1j}, D_{ij}\}$. Then $D_{11} + D_{ij} = D_{i1} + D_{1j}$. Therefore, all elements of D can be written as $D_{ij} = D_{i1} + (D_{1j} - D_{11}) = u_i + v_j$, where u and v are two sequences defined on $\{1, ..., N\}$. Since all row and column sums of D are the same, all elements of D are equal; therefore, all non-diagonal elements of C are equal, and finally $C = \hat{C}$. ∎

Therefore, we have finally established Proposition 8.

APPENDIX IV
LOAD-BALANCING GAIN (PROPOSITION 11)

We will now prove the equation in Proposition 11.

Proof: The right-hand-side of the equation comes directly from Proposition 10. The left-hand-side results from using the uniform mesh instead of the biased mesh to create the lower bound in Proposition 10. If $C_{uniform}$ is the uniform mesh, $\theta(C_{uniform}) = 1/2$ and $C_{uniform_{ij}} = 1/N$, therefore their ratio is $N/2$. ∎