On Scalability of Routing Tables in Dense Flat-Label Wireless Networks

Li-Yen Chen and Petar Momčilović

Abstract—**Consider a large wireless ad hoc network that facilitates communication between random pairs of network nodes. This paper investigates the size of routing tables as the number of nodes in the network increases. A routing protocol in a flat-label network is information-efficient if the amount of information at individual nodes required to route packets does not increase with the network size. It is shown that the shortest-path and straight-line routing algorithms are not information-efficient, i.e., these protocols can be implemented only when nodes' memory increases with the network size. On the other hand, it is established that there exists an information-efficient algorithm that routes packets correctly even if each node in the network is capable of storing information on a fixed number of destinations only.**

Index Terms—**Column-first routing, information-efficient, routing tables, scalability, wireless networks.**

I. Introduction

NETWORK scalability has emerged as a pivotal problem in designing architectures and protocols for large-scale multihop wireless networks. Indeed, large scales can significantly impact the performance of such networks since minor efficiencies, that can be tolerated in small networks, can accumulate and become a dominant factor determining the performance of large networks. Given that the growth of networking infrastructures is expected to continue in the future, it is of interest to contrive algorithms that support operation of large networks of nodes with limited hardware resources, e.g., memory, computational power, etc.

Most of scalability problems are impractical to be addressed experimentally due to a considerable cost of building large-scale prototypes. Moreover, even simulating such systems is often very difficult because of computational limitations. Mathematical study of large-scale wireless networks has been initiated in [1], where the authors established an upper bound on the network throughput as a function of the network size (number of nodes); the authors also constructed a protocol that achieves a slightly smaller throughput than the limiting one. Consequent studies focused on the throughput-delay trade-off [2], [3], achievability of the limiting throughput [4], and various extensions of the original model and their analyses [5]–[7]. The throughput capacity of wireless networks relates to that of lattice networks—see [8], [9] and the references therein.

Existing studies on the impact of limited node resources on the large-scale network performance are limited to results describing the dependency of the end-to-end throughput on the amount of available buffer space. In [9] the authors obtained an approximation of the throughput in a large finite-buffer lattice network with deterministic transmission times. A wireless network with exponential transmission times and finite-buffer nodes was considered in [10]. Buffer requirements in a wireless network with mobile nodes were discussed in [11].

It is recognized that design of scalable routing algorithms for wireless ad hoc networks is a difficult problem [12]. The difficulty is due to the fact that node identifiers (addresses) of adjacent nodes are not similar (flat-labeling), i.e., the address aggregation employed in the Internet is not directly applicable. Geographic routing [13]–[15] achieves aggregation by assigning additional identifiers to nodes based on their spacial location. However, we do not exploit this approach here, since in that case an additional mechanism is needed to provide translation between the original and location-based identifiers; resource requirements of such a DNS-like mechanism can be significant.

In this paper, we focus on one particular aspect of routing scalability. Namely, we consider the amount of information stored at network nodes that is required for routing packets between sources and respective destinations. We term a routing protocol information-efficient if it can operate correctly when each node stores information on a fixed number of destinations, regardless of the network size. That is, routing tables at network nodes do not increase with the network size. We show that the straight-line routing algorithm, considered in [1]–[3], and the closely related shortest-path algorithm are not information-efficient. However, we demonstrate that the column-first routing scheme is information-efficient. Such an algorithm induces routing tables that can be efficiently compressed.

The paper is organized as follows. In the next section we describe a wireless network model and state preliminary results. Routing tables are discussed in Section III. The following section contains scalability results on the straight-line and shortest-path routing algorithms. In Section V we demonstrate that there exists an information-efficient routing scheme for large wireless networks. A distributed algorithm is outlined in Section VI. Technical proofs can be found in Section VII.

Manuscript received July 21, 2008; revised July 23, 2010; accepted November 21, 2010. Date of current version June 22, 2011. This work was performed while the authors were with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA. This work was supported by the National Science Foundation under Grant CNS-0643213. The material in this paper was presented in part at the Annual IEEE Conference on Computer Communications, Anchorage, AK, May 2007.

L.-Y. Chen is with the network virtualization team at Microsoft Corp., Redmond, WA 98052 USA (e-mail: liche@microsoft.com).

P. Momčilović is with the Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: petar@ise.ufl.edu).

Communicated by A. Nosratinia, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2011.2146630

II. Model and Preliminaries

This paper considers a random topology network model with N nodes placed uniformly at random in a unit-area square. Each node is identified by a unique address (label). Node locations and addresses are independent, i.e., the Internet model of address aggregation is not applicable; such an assumption often holds for ad hoc networks. A node sends a packet to some other node in the network by specifying the destination node's address to the network layer, i.e., a packet contains its destination address.

We consider the relaxed protocol model of interference [2], [3]: a transmission from node i to node j is successful if for some $\delta > 0$ and all nodes $k \neq i$ transmitting simultaneously with i

$$\Delta(k,j) \geq (1+\delta)\Delta(i,j) \tag{1}$$

where $\Delta(i,j)$ is the Euclidian distance between nodes i and j. It is known [1] that, under reasonable assumptions, this model is equivalent to the popular physical model based on the signal-to-interference ratio.

We focus on a cell-based algorithm for packet transmissions. Specifically, the unit square is divided into square cells of area a_N, where the subscript indicates the network size. The shape of cells is square for the convenience of analysis; other ways of partitioning the unit square into cells can be considered, e.g., Voronoi tessellations [1] (see also Section VI). For simplicity, we assume that the quantity $1/\sqrt{a_N}$ is an integer so that all cells ($1/a_N$ in total) are of the same size and shape. Without loss of generality, let $\{a_N\}$ be a nonincreasing sequence in N. A packet is delivered from its source to its destination by forwarding between cells, i.e., between the nodes located in those cells. A packet can be forwarded from a cell to one of the four neighboring cells only (north, east, south, west); no multicell hops are permitted. Under the relaxed protocol model, the number of cells that interfere with any given cell is bounded by a constant independent of the network size N [2, Lemma 2]. In other words, a constant fraction of cells can transmit/receive packets simultaneously.

In order for the cell-based algorithm to deliver packets to their destinations, each cell that forwards packets should contain at least one node. This condition is satisfied by requiring that every cell in the unit square contains at least one node. To this end, let $k_N(i)$, $1 \leq i \leq 1/a_N$, be the number of nodes in the ith cell (the labeling of cells is arbitrary) when the network consists of N nodes. Given the uniform distribution of node locations, there exists a minimum cell size a_N that probabilistically ensures $k_N(i) > 0$ for all $1 \leq i \leq 1/a_N$. A stronger version of the following lemma was stated in [2]; it indicates that it is sufficient to have $a_N = \Omega(\ln N/N)$, as $N \to \infty$; we use the standard asymptotic notation, see e.g., [16, Section I.3].

Lemma 1: If $a_N > a \ln N/N$ for some $a > 1$ then, as $N \to \infty$

$$\mathbb{P}\left[\min_{1\leq i\leq 1/a_N} k_N(i) = 0\right] \to 0.$$

The next lemma provides an additional characterization of the minimum and maximum number of nodes in a single cell. Note that the expected number of nodes in a cell is given by $\mathbb{E}k_N(i) = a_N N$.

Lemma 2: Suppose that $a_N > a\ln N/N$ for some $a > 1$. There exist $0 < \gamma_1 < \gamma_2 < \infty$ such that, as $N \to \infty$

$$\mathbb{P}\left[\min_{1\leq i\leq 1/a_N} k_N(i) \geq \gamma_1 a_N N\right] \to 1 \tag{2}$$

and

$$\mathbb{P}\left[\max_{1\leq i\leq 1/a_N} k_N(i) \leq \gamma_2 a_N N\right] \to 1. \tag{3}$$

Proof: See Section VII. ■

Next, we describe the configuration of traffic patterns. Each node selects a *point* on the unit square uniformly at random as its destination; all destinations are selected independently. Since there will be no node at a chosen point with probability 1, packets from the corresponding source will be delivered to one of the nodes that share the cell with its destination point. A similar assumption was made in [1]—it ensures the independence of traffic routes. A routing algorithm specifies a set of cells that forward packets between sources and respective destinations. It this paper, we consider three different routing policies: straight-line, shortest-path and column-first. These policies operate as follows:

- *Straight-line:* A source node and destination point are connected by the straight line. Packets are forwarded along the cells intersected by this line. This policy was employed in [1]–[3], because it facilitates mathematical analyses. However, it cannot be implemented easily in practice without sophisticated node hardware and/or central authority.
- *Shortest-path:* A route is selected in such a way that it contains the minimum possible number of cells (hops). When a cell can achieve the same number of hops via multiple neighbors, the cell chooses one of these neighbors uniformly at random. The main advantage of this policy is that it can be implemented via the distributed asynchronous Bellman-Ford algorithm [17, Section 5.2].
- *Column-first:* Packets are delivered from a source cell to a destination cell in two phases [18, Section 1.7]. First, they are forwarded along the column that contains the source cell until they reach the row that contains the destination cell. In the second phase, packets are forwarded along the row to their destination.

We conclude the description of the routing policies with an observation that, for the considered model (unit square and square cells), the three algorithms produce routes of the same hop (cell) length for any source-destination pair.

One more aspect of the network model needs to be specified: the amount of time required to correctly transmit a packet between two neighboring cells. Packet transmission times can be modeled as either deterministic [1]–[3], [9] or stochastic [10] quantities, e.g., in [10] it was assumed that the transmission times are exponentially distributed. The expected transmission time is a function of the packet size in bits. Since we assume that destination addresses are included in each packet's meta-data field, the packet size has to increase at least logarithmically in the network size N, i.e., $\lceil \log_2 N \rceil$ bits are needed to uniquely identify all nodes in the network.

For simplicity, we assume that each node has an infinite amount of buffer space to store packets temporarily in the course of the forwarding process. The network throughput ϑ_N is feasible if *all* N source-destination pairs can communicate at the long-term rate ϑ_N. The cell-based algorithm is optimal in the sense that it achieves (asymptotically) the optimal trade-off between the throughput and delay [2], [3]. Studies [2], [3] assumed deterministic service times with a time-slotted transmission system and employed straight-line routing. It is straightforward to extend Theorem 1 of [2] for the case of the shortest-path and column-first routing policies as follows.

Theorem 1: Consider network nodes capable of storing arbitrary amounts of routing information. Suppose that straight-line, shortest-path or column-first routing is employed. Let $a_N > a \ln N/N$ for some $a > 1$. The network throughput scales as $\vartheta_N = \Theta(a_N^{-1/2}/N)$, i.e., there exist constants $0 < \zeta_1 < \zeta_2 < \infty$ such that

$$\mathbb{P}\left[\vartheta_N < \frac{\zeta_1}{\sqrt{a_N}N} \text{ is feasible}\right] \to 1$$

as $N \to \infty$, and

$$\liminf_{N\to\infty} \mathbb{P}\left[\vartheta_N > \frac{\zeta_2}{\sqrt{a_N}N} \text{ is feasible}\right] < 1.$$

Since the throughput scaling is better understood for the model with deterministic transmission times and time-slotted system, we keep the discussion on the trade-off between the throughput and size of routing tables restricted to that model (see Corollary 1). However, we point out that Theorem 1 can be extended to cover the model with exponentially distributed transmission times (a time-slotted system is not applicable in that case). Namely, it can be shown that the network is equivalent to a (queueing) Kelly network [19, Chapter 3] for which stability results are available. When the transmission times are random but not exponential, the network stability can depend on the routing policy and can be hard to evaluate in general.

III. Routing Tables

In this section, we discuss the size of routing tables at individual nodes required for the correct operation of routing protocols. We say that a routing algorithm operates correctly if packets from all sources can be delivered to their respective destinations in the cell-based algorithm.

A. Types of Routing Tables

Routing tables contain *records*, i.e., the information on how packets to specific destinations need to be handled. The amount of information required for correct routing depends on the type of a record, i.e., whether packets contain additional meta-data besides destination addresses. Two cases are of a particular interest:

- *Destination address only*. Due to the broadcast nature of packet transmissions in wireless networks, a packet can be received by all nodes that satisfy the interference constraint (1). Each node keeps a list of destination addresses it forwards packets to. A node examines all packets it receives and either retransmits them (at a later time) if they are on the list, or simply discards them. In this case, a routing table consists of a list of destination addresses. A mechanism is needed to prevent a node from transmitting the same packet multiple times.
- *Destination address & adjacent cell identifiers*. All cells are assigned labels such that each cell can uniquely identify all of its interfering cells. Each packet contains cell labels of the transmitting and receiving nodes, i.e., a pair (η, ω); this is in addition to the address of the destination node. Nodes maintain (possibly in compressed form) lists of triples of the form $(\alpha,\, \iota,\, \omega)$, where α is the address of a destination node, ι is an input cell and ω is an output cell. A node is responsible for forwarding packets with destination α from cell ι to cell ω (hence, the term input and output cells). A node receives only packets that contain address α and transmitting-cell label ι such that $(\alpha, \iota, \cdot)$ is on the node's list. Before retransmitting a packet, a node updates packet's meta-data field with its own cell label and the receiving-cell label ω from the triple (α, ι, ω). We observe that, in this case, a cell can be treated as a switch/router with neighboring cells being input/output ports.

We focus on the second type of records for two reasons. First, compressing routing tables in the former case is difficult because the randomness of address locations results in high entropy of the list; extra information (cell labels) helps with compression. Second, the overhead due to cell labels is small. Since each cell has a limited number c of interfering cells (see Section II), it is sufficient to have $(c+1)$ labels only in order to have each cell identify its interfering cells uniquely [16, p. 1092], i.e., $\lceil \log_2(c+1) \rceil$ bits per cell label are sufficient (which is negligible compared to $\lceil \log_2 N \rceil$ bits size of α). In addition, such labels are available at a very little cost due to the layered architecture of current networks, e.g., one can exploit the already existing layer-2 addressing space. Finally, we remark that one can also design routing tables with other types of records. For example, records of type (α, ω) are feasible, i.e., the input cell identifier is eliminated in the triple (α, ι, ω). However, such a design does not substantially decrease the record size in bits, while the ability to compress routing tables is compromised.

B. Size of Routing Tables

Next, we turn our attention to the size of routing tables. Due to our model (flat-labeling), a prefix-based compression [20] of routing tables is not applicable. Let $C_N(i)$ be the set of routing records in cell i needed to correctly forward packets between sources and destinations. All the information required in a cell has to be stored at nodes located in that cell, i.e.,

$$C_N(i) = \bigcup_{n \in i} T_N(n)$$

where $T_N(n)$ is the set of records stored at node n. By $\|C_N(i)\|$ and $\|T_N(n)\|$ we denote the sizes of routing tables $C_N(i)$ and $T_N(n)$, respectively, measured in number of address length, i.e., if $\|C_N(i)\| = x$ then the size of compressed $C_N(i)$ in bits is given by $x\lceil \log_2 N \rceil$. Effectively, we assume that each node can hold a packet in its memory (since the packet size is lower bounded by $\lfloor \log_2 N \rfloor$ bits). Given that all nodes have identical capabilities (hardware), it is reasonable to assume that the routing information is spread evenly among nodes in a cell. This implies, for all $i \in n$

$$\|T_N(n)\| \geq \lfloor \|C_N(i)\|/k_N(i) \rfloor \tag{4}$$

(recall that $k_N(i)$ is the number of nodes in the ith cell). Data stored at each node should contain not only a compressed routing table, but also an decompression algorithm. Thus, it would perhaps be more appropriate to consider the Kolmogorov-Chaitin complexity [21, Chapter 7] of the routing table $C_N(i)$. However, estimating the Kolmogorov-Chaitin complexity of the routing table is difficult in general. Therefore, we establish a lower bound by considering the traditional information-theoretic compression. In that case, the compressed data do not have to contain a decompression algorithm (which is not true in the case of the Kolmogorov-Chaitin complexity). Note that considering specific algorithms for compression of general tables (such as in [22]) will not provide a lower bound on $\|C_N(i)\|$.

Let $\psi_d(i)$ be the number of flows forwarded by cell i in direction d. Direction d indicates the neighbors of cell i between which it forwards packets (i.e., the input and output cells), e.g., south-to-north, north-to-east, etc. In other words, for a given cell, direction d can be represented by a pair of neighboring cell labels (ι, ω) with $\iota \neq \omega$. For all nonedge cells in the unit-area square, d can take 12 possible values; denote the set of these values by $\mathcal{D}$.

A routing table $C_N(i)$ can be thought of as a discrete-valued function R that has two inputs and one output

$$\omega = R(\alpha, \iota) \tag{5}$$

where ω is the output cell label computed based on the destination address α and the input-cell label ι; for compression of functions in communication contexts see [23], [24].

The following lemma establishes a lower bound on the amount of routing information stored at individual nodes. The bound is based on the minimum value of $\psi_d(i)$ over $d \in \mathcal{D}$. The intuition behind the lemma is as follows. For a given cell i, consider the function $R(\cdot, \iota)$ for some specific ι. Let α be a destination address (flow) chosen uniformly at random among flows that have packets forwarded by cells ι and i. Then the "randomness" of $R(\alpha, \iota)$ is determined by $\psi_d(i)$ for $d = (\iota, \cdot)$, since $\psi_d(i)$, for $d = (\iota, \cdot)$, determines the likelihood of possible values of $R(\alpha, \cdot)$. The higher the randomness of $R(\alpha, \iota)$ (or entropy of $R(\cdot, \iota)$) the more information is need to correctly route packets. In the extreme case, when $R(\alpha, \iota)$ is deterministic, no information about individual destination needs to be stored—routing can be performed solely based on the input cell labels (in this case, all packets arriving from input cell ι are forwarded to a single neighboring cell regardless of the destination address α).

Lemma 3: Consider a scheme that operates correctly for all realizations of address locations. Let $\psi := \min_{d \in \mathcal{D}} \psi_d(i)$. There exists $\zeta > 0$ such that for any node n in cell i, we have

$$\|T_N(n)\| \geq \frac{\zeta \psi}{k_N(i) \ln N}.$$

Proof: For notational simplicity let $\hat{\psi} := \sum_{d \in \mathcal{D}} \psi_d(i)$. It is sufficient to consider the case $\psi > 0$ only; otherwise the lemma holds trivially. The proof is based on a counting argument. For $d = (\iota, \omega)$, define

$$A_d(i) := \{\alpha : \omega = R(\alpha, \iota)\} \tag{6}$$

as the set of destination addresses forwarded by cell i in direction d. The minimum number of *bits* required to describe $R(\cdot, \cdot)$ is lower bounded by the number of bits required to describe $A_d(i)$ with the smallest cardinality (across $d \in \mathcal{D}$) so that $\{\alpha \in A_d(i)\}$ is verifiable. This number of bits is further lower bounded by $\log_2 \binom{\hat{\psi}}{\psi}$, since ψ is the cardinality of the smallest of sets $A_d(i)$, $d \in \mathcal{D}$, and $\hat{\psi}$ is the total number of destination addresses forwarded by cell i; for a given set of cardinality $\hat{\psi}$, the number of subsets with cardinality ψ is given by $\binom{\hat{\psi}}{\psi}$. Moreover, for $\psi > 0$

$$\binom{\hat{\psi}}{\psi} \geq \frac{(\hat{\psi} - \psi)^{\psi}}{\psi!} \geq (\hat{\psi}/\psi - 1)^{\psi}$$

and hence, there exists $\zeta > 0$, such that

$$\|C_N(i)\| \geq \zeta \psi / \ln N$$

we used the fact the address length is equal to $\lceil \log_2 N \rceil$ bits. Recalling (4) completes the proof. ∎

C. Compression

In this subsection, we discuss a practical compression scheme that, in certain cases, can achieve performance that is close to the lower bound from Lemma 3. The algorithm exploits the difference in cardinalities of sets $A_d(i)$ for fixed i and different d's (see (6) for the definition), i.e., the compression is based on the existence of dominant routing directions. Consider a forwarding cell i and one of its neighbors ι. A single-argument routing/forwarding function $R(\cdot, \iota)$ (see (5)) can be expressed in the following form:

$$R(\alpha, \iota) = \begin{cases} \omega_2, & \text{if } \alpha \in A_{d_2}(i), d_2 = (\iota, \omega_2) \\ \omega_3, & \text{if } \alpha \in A_{d_3}(i), d_3 = (\iota, \omega_3) \\ \omega_1, & \text{otherwise} \end{cases} \tag{7}$$

where $\iota \neq \omega_1 \neq \omega_2 \neq \omega_3$ are the four neighbors of cell i. Then, the function $R(\cdot, \iota)$ can be described by the two sets $A_{d_2}(i)$ and $A_{d_3}(i)$. Upon a reception of a packet, cell i examines whether the destination address α belongs to either A_{d_2} or A_{d_3}. If it does, the packet is forwarded to either cell ω_2 or cell ω_3; otherwise to cell ω_1. The number of bits required to describe $R(\cdot, \iota)$ in the form (7) is upper bounded by

$$(\|A_{d_2}(i)\| + \|A_{d_3}(i)\|)\,(\lceil \log_2 N \rceil + 1)$$

where $\|A_d(i)\|$ is the number of destination addresses in set $A_d(i)$. Selecting d_1 in such a way that it satisfies

$$A_{d_1}(i) = \max_{d \in \{d_1, d_2, d_3\}} \|A_d(i)\|$$

address		output cell
α_3	0010	1
α_4	0011	1
α_5	0100	1
α_6	0101	2
α_7	0110	3
α_8	0111	3
α_9	1000	1
α_{10}	1001	1
α_{11}	1010	2
α_{12}	1011	3
α_{13}	1100	1
α_{15}	1110	2
α_{16}	1111	1

prefix	output cell
00	1
0100	1
0101	2
011	3
100	1
1010	2
1011	3
110	1
1110	2
1111	1

prefix	output cell
*	1
0101	2
011	3
1010	2
1011	3
1110	2

Fig. 1. Example of $R(\cdot,\iota)$ for a fixed ι and its reduced form by applying longest-prefix compression. The table on the left (corresponding to the tree representation on the top in Fig. 2) is the routing table before compression; the table in the center corresponds to the tree on the bottom in Fig. 2, which is the routing table after the merging procedure; on the right is a fully compressed version of the routing table.

results in the minimum number of bits required (for this particular scheme). The algorithm is particularly effective when $\|A_{d_1}(i)\| \gg \|A_{d_2}(i)\| + \|A_{d_3}(i)\|$. To this end, for each input node ι, the largest set $A_{d_1}(i)$ can be eliminated from the routing table. Instead, a packet from ι is directed according to d_1 by default, if it carries an destination address that does not belong to either $A_{d_2}(i)$ or $A_{d_3}(i)$. We further exploit this idea in Section V.

D. Example: Longest-Prefix Compression

The idea of biasing the distribution of routing directions in order to reduce the size of forwarding tables is not limited to a specific compression scheme employed. In the following, we illustrate this point by examining the well-known longest-prefix compression algorithm [25]. Namely, we compare the performance of this compression scheme under various cardinalities of sets A_{d_i}'s (e.g., with or without dominant forwarding directions). For simplicity, suppose that packets to a given destination are forwarded by cell c from cell ι in the direction d_i with probability p_i for $1 \le i \le 3$. In a given cell, forwarding directions for different destinations are independent. Thus, the total number flows forwarded by cell c from cell ι is binomially distributed with parameters (N,p), where $p = p_1 + p_2 + p_3$ is the probability that a given destination address has its packets forwarded by cell c from cell ι. An example of an uncompressed routing table is shown on the left in Fig. 1. Upon receiving a packet from cell ι, cell c looks up the destination address in the table and forwards the packet to the appropriate output cell (indexed by 1, 2, and 3).

Next, we describe a binary tree representation of the routing table in order to illustrate longest-prefix compression. For convenience, assume that the number of nodes N is a power of 2; addresses in the network are one-to-one mapped to the leaves of a complete binary tree of depth $\log_2 N$. For each address $\alpha \in \{0,1\}^{\log_2 N}$ we store the output cell identifier at the leaf node corresponding to α as shown on the top in Fig. 2. If packets to destination α are not forwarded by cell c from cell ι, then the leaf node corresponding to α is empty. We say that a set of leaf nodes have nonconflicting indices if they do not contain more than one cell index. Applying longest-prefix compression to the set of addresses is equivalent to iterating the following merging procedure on the tree. If a leaf and its sibling have nonconflicting indices, they are merged into a new node (either empty or with an appropriate cell index). We iterate the procedure until no more leaves can be merged. An example of a tree before and after this operation is shown in Fig. 2; the corresponding routing tables can be found in Fig. 1. After the merging process, each leaf corresponds to a prefix stored in the routing table. Note that node x in the *binary tree* is stored in the routing table if and only if (i) all leaf descendants of x have nonconflicting indices with at least one leaf being nonempty, and (ii) there is a pair of leaf descendants of the parent of x such that they have conflicting indices. Observe that one of the output cell indices can be removed from the routing table without producing ambiguity. Without loss of generality, assume that $p_1 \ge \max\{p_2,p_3\}$. Then, prefixes with output cell indices 1 can be replaced by a single entry $(*,1)$ in the table, as shown on the right in Fig. 1 (the symbol "$*$" represents a zero-length prefix). Whenever a packet arrives, it is forwarded to an output cell according to the longest-prefix in the routing table that matches the destination address.

We now estimate the expected number of bits needed to store all the prefixes. Clearly, the performance of longest-prefix compression depends on the content stored at each leaf. Define B_x^j as the event that children of x can be merged to a node with index j, for $1 \le j \le 3$, and B_x^0 as the event that all descendant leaves of x are empty. Then, if node x is of height $l(x) \in \{0,1,\ldots,\log_2 N\}$, the probabilities of B_x^j are as follows:

$$\mathbb{P}\left[B_x^j\right] = \begin{cases} p_0^{2^{l(x)}}, \text{if } j=0 \\ (p_0+p_j)^{2^{l(x)}} - p_0^{2^{l(x)}}, \text{if } 1 \le j \le 3 \end{cases} \quad (8)$$

where $p_0 = 1-p$ is the probability that a leaf node is empty. Let B_x be the event that x is a prefix leaf with cell index $j \ne 1$ (i.e., a leaf in the compressed (merged) tree with cell index $j \ne 1$). This event indicates that the prefix corresponding to x is stored in the compressed version of routing table. Then

$$B_x = \bigcup_{j\in\{2,3\}} \left\{ B_x^j \cap \overline{\left\{ B_{\sigma(x)}^j \cup B_{\sigma(x)}^0 \right\}} \right\}$$

where $\sigma(x)$ is the sibling of x in the original tree, and $\overline{B}$ denotes the complement of B. Given the height $l(x)$ of x, the conditional probability of event B_x, due to (8), satisfies

$$\begin{aligned} &\mathbb{P}[B_x \mid l(x)=i] \\ &\quad = \sum_{j\in\{2,3\}} \mathbb{P}\left[B_x^j\right] \left(1 - \mathbb{P}\left[B_{\sigma(x)}^j \cup B_{\sigma(x)}^0\right]\right) \\ &\quad = \sum_{j\in\{2,3\}} \left((p_0+p_j)^{2^i} - p_0^{2^i}\right)\left(1-(p_0+p_j)^{2^i}\right). \quad (9) \end{aligned}$$

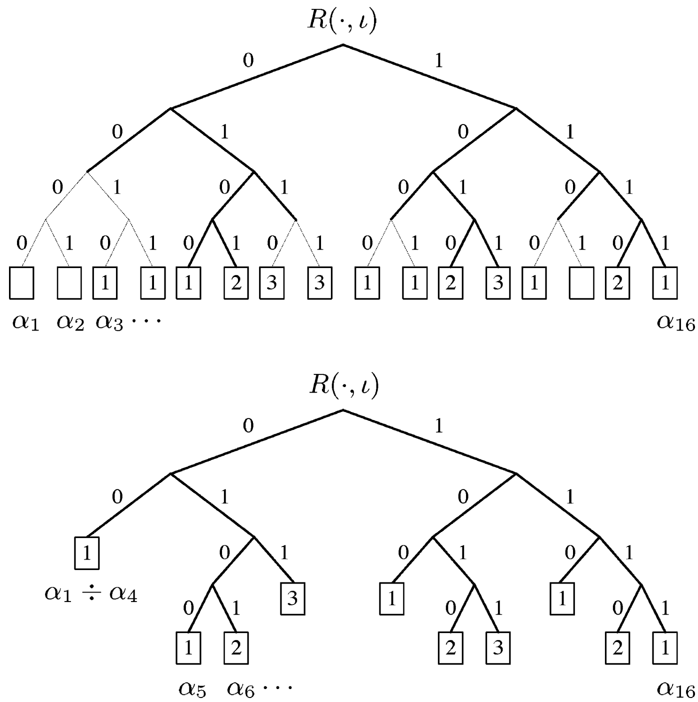

Fig. 2. Example of $R(\cdot, \iota)$ in Fig. 1, represented by a binary tree. Each leaf of the tree corresponds to an address of length $\log_2 N$ bits, specified by the unique path from the root. The tree can be compressed by merging leaves with nonconflicting indices.

The expected number of bits needed (under longest-prefix compression) for the routing table stored in cell c is given by

$$C := \mathbb{E}\left[\sum_{i=0}^{\log_2 N} \sum_{x:l(x)=i} (\log_2 N - i) \cdot 1_{\{B_x\}}\right]$$
$$= \sum_{i=0}^{\log_2 N} 2^{\log_2 N - i}(\log_2 N - i)\, \mathbb{P}[B_x \mid l(x) = i]. \quad (10)$$

Next, we examine two contrasting cases that illustrate the impact of routing directions on the size of routing tables:

- *Uniform forwarding directions*: $p_1 = p_2 = p_3 = p/3$. From (9), it follows that

$$\mathbb{P}[B_x \mid l(x) = i] = 2\left((1-2p/3)^{2^i} - (1-p)^{2^i}\right) \times \left(1 - (1-2p/3)^{2^i}\right). \quad (11)$$

  Note that a throughput-efficient routing algorithm requires that $p = p(N) \to 0$ as $N \to \infty$, i.e., each node forwards packets for only a negligible fraction of all nodes. The summation in (10) can be lower bounded by a single term corresponding to $i = \lfloor -\log_2 p \rfloor$; in that case, $\mathbb{P}[B_x \mid l(x) = i]$ is $\Theta(1)$, as $N \to \infty$. Thus, (10) and (11) yield

$$C \geq Np\ \log_2(Np)\ \mathbb{P}[B_x \mid l(x) = \lfloor -\log_2 p \rfloor]$$
$$= c_1(Np \ln N)(1 + o(1)) \quad (12)$$

  as $N \to \infty$, for some constant $c_1 > 0$. The last equality is due to the fact that p must dominate $O(1/N)$ as $N \to \infty$; indeed, a packet with a given destination address is typically forwarded by at most a negligible fraction of nodes in the network. Note that the expected number of bits needed to store the routing table before compression is $Np \log_2 N$, which is of the same order as the right-hand side of (12) (empty leaf nodes need not be stored). Hence, when the distribution of forwarding directions is uniform, the longest-prefix algorithm results in a constant factor reduction only of the size of routing tables.
- *Dominant forwarding direction*: $p_1 \gg p_2 = p_3$. Note that this corresponds to $\|A_{d_1}\| \gg \|A_{d_2}\| + \|A_{d_3}\|$ (see the previous subsection).
  In this case, the conditional probability in (9) evaluates to

$$\mathbb{P}[B_x \mid l(x) = i] = 2\left((1-p+p_2)^{2^i} - (1-p)^{2^i}\right) \times \left(1 - (1-p+p_2)^{2^i}\right).$$

  Thus, as $N \to \infty$, the expected number of bits needed for the routing table is upper bounded by

$$C \leq \log_2 N\, (Np \log_2(Np)\mathbb{P}[B_x \mid l(x) = \lceil -\log_2 p \rceil])$$
$$= c_2\, \frac{p_2 \ln N}{p_1}\ (Np \ln N)\, (1 + o(1))$$

  as $N \to \infty$, for some constant $c_2 < \infty$. That is, if $p_1 \gg p_2 \ln N$, then the size of routing table is compressed with reduction of factor proportional to $p_1/(p_2 \ln N) \gg 1$. Therefore, the more biasing the dominant forwarding direction is, the better the performance of the longest-prefix compression.

IV. Shortest-Path Routing

In this section, we focus on two routing algorithms: straight-line and shortest-path. The straight-line algorithm is determin-

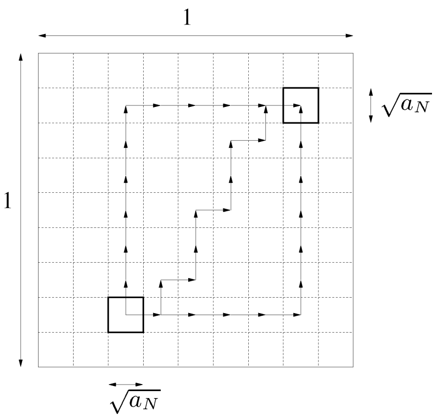

Fig. 3. There exist multiple shortest paths (in the number of hops) between two cells located in different rows and columns. Three different paths are shown in this example. When a cell can achieve the same number of hops to the destination cell via 2 different neighbors, it chooses one at random.

istic, i.e., for a fixed pair of cells it always produces the same route; the shortest-path algorithm is randomized.

In addition to referring to cells by a single index $i \in \{1,\ldots,N\}$, we also refer to cells by a pair of indices (i,j), $i,j \in \{1,\ldots,1/\sqrt{a_N}\}$ when convenient. These indices specify the relative position of cells with respect to the lower-left corner of the unit square. The cells in the lower-left and upper-right corners are labeled by (1,1) and $(1/\sqrt{a_N}, 1/\sqrt{a_N})$, respectively. When cell locations are of significance, we will use the 2-D indexing.

Consider a pair of source-destination cells as shown in Fig. 3. The shortest path (in the number of hops) between these cells is not unique. In fact, given that the source and destination nodes are located in cells (m,n) and (u,v), respectively, the number of shortest paths is given by $\binom{|u-m|+|v-n|}{|u-m|}$; each of these paths is $|u-m|+|v-n|$ hops long. Unless the source and destination cells are located in the same row or column, the destination cell can be reached via two different neighbors of the source. Given that the sole goal is to minimize the number of hops, neither of the neighbors is preferred over the other one. Thus, our assumption that cells choose uniformly at random between the alternatives (neighbors). Furthermore, in the absence of geographic information at individual cells, random choice of neighbors appears to be a natural solution.

The next lemma establishes a lower bound on the number of forwarded flows in all directions by a large fraction of cells in the network. This result coupled with the fact that most cells forward packets to $\Theta(\sqrt{a_N}N)$ destinations (because a typical distance between a source and destination is $\Theta(1/\sqrt{a_N})$ cells) implies that forwarded flows are spread evenly among the elements of set $\mathcal{D}$ (modulo a constant prefactor). For a set of nodes or cells $\mathcal{S}$, let $\|\mathcal{S}\|$ be the cardinality of the set.

Lemma 4: Consider either straight-line or shortest-path routing. Let $a_N \geq 1/N$ such that $a_N \to 0$ as $N \to \infty$. For arbitrary $\delta \in (0,1)$ there exists a sequence of sets $\mathcal{C}_N \subseteq \{1,2,\ldots,a_N^{-1}\}$ such that $a_N\|\mathcal{C}_N\| \geq 1-\delta$ for all N large enough and

$$N\mathbb{P}[\psi_d(i) < \varepsilon\sqrt{a_N}N] \to 0$$

as $N \to \infty$, for some $\varepsilon > 0$, all $i \in \mathcal{C}_N$ and all $d \in \mathcal{D}$.

Proof: See Section VII. ■

Remark 1: The conditions of the lemma on the cell size are not restrictive. First, the case $a_N < 1/N$ is not relevant to the cell-based algorithm in view of Lemma 1. Second, when $\lim a_N > 0$, only $\Theta(1/N)$ throughput per source-destination pair can be achieved, cf. Theorem 1; however, $\Theta(1/N)$ throughput can be achieved without multihop forwarding.

The following theorem is the first main result of the paper. It characterizes the size of routing tables at individual nodes under straight-line and shortest-path routing. Effectively, the theorem states that within our model these algorithms are not scalable in the sense that they require excessively large routing tables when throughputs close to the limiting one are desired ($a_N = o(\ln^{-2} N)$ as $N \to \infty$, cf. Fig. 3).

Theorem 2: Consider either straight-line or shortest-path routing. Let $a_N > a\ln N/N$, for some $a > 1$, such that $a_N \to 0$ as $N \to \infty$. For arbitrary $\delta \in (0,1)$, routing tables at $(1-\delta)N$ nodes scale as $\Omega(a_N^{-1/2}/\ln N)$ as $N \to \infty$. Namely, there exists a sequence of sets $\mathcal{N}_N \subseteq \{1,2,\ldots,N\}$ with $\|\mathcal{N}_N\| \geq (1-\delta)N$, and $\varepsilon > 0$ such that, as $N \to \infty$

$$\mathbb{P}\left[\min_{n\in\mathcal{N}_N} \|T_N(n)\| \geq \frac{\varepsilon}{\sqrt{a_N}\ln N}\right] \to 1$$

i.e., when $a_N = o(\ln^{-2} N)$ as $N \to \infty$, the size of the routing tables increases without a bound as the network size increases.

Remark 2: The theorem also holds for a wide set of network topologies that satisfy Lemma 4. Effectively, the theorem holds for topologies under which there is no significant correlation between input and output cells (see Section III).

Proof: We start with obtaining an estimate on the number of nodes that forward a large number of flows in all 12 possible directions. To this end, let $\mathcal{C}_N$ be a set of cells with $\|\mathcal{C}_N\| \geq (1-\zeta)a_N^{-1}$ for some $\zeta \in (0,1)$; recall that the total number of cells is a_N^{-1}. It is appropriate to think of $\mathcal{C}_N$ as the set of cells that are not "too close" to the edges of the unit square. De Morgan's law and the union bound yield

$$\begin{aligned}
&\mathbb{P}\left[\min_{i\in\mathcal{C}_N}\min_{d\in\mathcal{D}}\psi_d(i) \geq \xi\sqrt{a_N}N\right] \\
&\quad= \mathbb{P}\left[\bigcap_{i\in\mathcal{C}_N}\bigcap_{d\in\mathcal{D}}\{\psi_d(i) \geq \xi\sqrt{a_N}N\}\right] \\
&\quad= 1-\mathbb{P}\left[\bigcup_{i\in\mathcal{C}_N}\bigcup_{d\in\mathcal{D}}\{\psi_d(i) < \xi\sqrt{a_N}N\}\right] \\
&\quad\geq 1-\sum_{i\in\mathcal{C}_N}\sum_{d\in\mathcal{D}}\mathbb{P}\left[\psi_d(i) < \xi\sqrt{a_N}N\right] \\
&\quad\geq 1-\|\mathcal{D}\|\|\mathcal{C}_N\|\max_{i\in\mathcal{C}_N}\max_{d\in\mathcal{D}}\mathbb{P}[\psi_d(i) < \xi\sqrt{a_N}N] \\
&\quad> 1-12N\max_{i\in\mathcal{C}_N}\max_{d\in\mathcal{D}}\mathbb{P}[\psi_d(i) < \xi\sqrt{a_N}N]
\end{aligned}$$

where the last inequality follows from $\|\mathcal{D}\| = 12$ and $\|\mathcal{C}_N\| < N$. The preceding inequality and Lemma 4 imply that there exists a set of cells $\mathcal{C}_N$ which contains $(1-\zeta)$ fraction of cells such that, as $N \to \infty$,

$$\mathbb{P}\left[\min_{i\in\mathcal{C}_N}\min_{d\in\mathcal{D}}\psi_d(i) \geq \xi\sqrt{a_N}N\right] \to 1 \qquad (13)$$

i.e., $(1-\zeta)$ fraction of cells forward at least $\xi\sqrt{a_N}N$ flows in *all* directions, where $\xi > 0$.

Now, consider a cell $i \in \mathcal{C}_N$. According to Lemma 3, the following bound holds for all nodes n located within cell i

$$\|T_N(n)\| \geq \zeta \frac{\min_{d\in\mathcal{D}} \psi_d(i)}{k_N(i)\ln N} \tag{14}$$

for some $\zeta > 0$. Define set $\mathcal{N}_N$ of nodes located within cells in set $\mathcal{C}_N$

$$\mathcal{N}_N := \{n \in \{1,2,\ldots,N\} : n \in i \in \mathcal{C}_N\}$$

and note that by the strong law of large numbers

$$\liminf_{N\to\infty} \frac{\|\mathcal{N}_N\|}{N} > 1-\delta \quad \text{a.s.} \tag{15}$$

where $\delta \in (0,1)$ is such that $\delta \to 0$ as $\zeta \to 0$. By applying the minimum operator on both sides of (14), we obtain a lower bound on the size of routing tables at all nodes in $\mathcal{N}_N$

$$\begin{aligned}\min_{n\in\mathcal{N}_N} \|T_N(n)\| &\geq \zeta \min_{n\in\mathcal{N}_N} \left\{\frac{\min_{d\in\mathcal{D}} \psi_d(i(n))}{k_N(i(n))\ln N}\right\}\\ &= \zeta \min_{i\in\mathcal{C}_N} \left\{\frac{\min_{d\in\mathcal{D}} \psi_d(i)}{k_N(i)\ln N}\right\}\\ &\geq \zeta \frac{\min_{i\in\mathcal{C}_N}\min_{d\in\mathcal{D}} \psi_d(i)}{\max_{i\in\mathcal{C}_N} k_N(i)\ln N}\end{aligned}$$

where $i(n)$ is the cell that contains node n. Finally, the preceding inequality, (13), Lemma 2 and (15) imply the statement of the theorem. ■

Theorem 2 allows one to quantify the required memory at network elements for a given cell tessellation parameter a_N. Since this parameter a_N also determines the throughput for the model with deterministic transmission times (see Theorem 1), one concludes that the amount of available memory at individual nodes limits the maximum throughput. The following corollary establishes the dependency of the throughput on the memory size at individual nodes.

Corollary 1: Consider either shortest-path or straight-line routing in a network with $a_N > a\ln N/N$ for some $a > 1$. Let each node be capable of storing b_N addresses in its routing table when the network consists of N nodes. If $b_N = O(\sqrt{N/\ln N})$ as $N\to\infty$, then there exist $0 < \zeta_1 < \zeta_2 < \infty$ such that

$$\mathbb{P}[\vartheta_N < \zeta_1 b_N/N \text{ is feasible}] \to 1 \tag{16}$$

as $N\to\infty$, and

$$\liminf_{N\to\infty} \mathbb{P}\left[\vartheta_N > \zeta_2 b_N \ln N/N \text{ is feasible}\right] < 1. \tag{17}$$

Remark 3: If $b_N = \Theta(\sqrt{N/\ln N})$ then the ratio that determines ϑ_N satisfies $b_N/N = \Theta(1/\sqrt{N\ln N})$. Given that the maximum achievable throughput under the cell-based algorithm is $\Theta(1/\sqrt{N\ln N})$, increasing b_N beyond $\Theta(\sqrt{N/\ln N})$ does not asymptotically improve the throughput.

Proof: We prove (16) first. Set $\sqrt{a_N} = \lceil c/b_N\rceil$ for some constant $c > 0$. Due to the assumption $b_N = O(\sqrt{N/\ln N})$ it is possible to select c such that $a_N > a\ln N/N$ with $a > 1$. Hence, there exists $\xi_1 > 0$ such that with no limit on the size of routing tables we have

$$\mathbb{P}\left[\vartheta_N < \frac{\xi_1}{\sqrt{a_N}N} \text{ is feasible}\right] \to 1 \tag{18}$$

as $N\to\infty$. Next we establish an upper bound on the size of a routing tables in all cells. The size of the routing table in cell i is upper bounded by the number of flows traversing the cell, $\sum_{d\in\mathcal{D}}\psi_d(i)$. If throughput ϑ_N is achievable, then there exists a fixed $\delta > 0$ such that for all $1 \leq i \leq 1/a_N$

$$\sum_{d\in\mathcal{D}} \psi_d(i) < \delta/\vartheta_N$$

where δ can be interpreted as the maximum transmission rate of a cell. Limit (18) and the preceding inequality yield, as $N\to\infty$,

$$\mathbb{P}\left[\max_{1\leq i\leq 1/a_N} \sum_{d\in\mathcal{D}} \psi_d(i) < \delta\sqrt{a_N}N/\xi_2\right] \to 1 \tag{19}$$

where $\xi_2 < \xi_1$. On the other hand, the minimum number of routing records that can be stored in a cell can be estimated by Lemma 2

$$\mathbb{P}\left[b_N \min_{1\leq i\leq 1/a_N} k_N(i) > \varepsilon c\sqrt{a_N}N\right] \to 1 \tag{20}$$

as $N\to\infty$, for some $\varepsilon > 0$; we also used the relationship between a_N and b_N. Combining (19) and (20) yields that, for c large enough, each cell has enough memory space to store all the required routing records in the limit as $N\to\infty$

$$\mathbb{P}\left[\max_{1\leq i\leq 1/a_N} \sum_{d\in\mathcal{D}} \psi_d(i) < b_N \min_{1\leq i\leq 1/a_N} k_N(i)\right] \to 1.$$

Therefore, (18) holds also for the case when each node can store b_N addresses rather than an infinite number of them. Recalling that $\sqrt{a_N} = \lceil c/b_N\rceil$ renders (16).

Now we consider (17). Note that if $b_N = 0$, then there is no memory available for routing tables at all, and thus, one requires $b_N \geq 1$. It suffices to consider the case $a_N \to 0$ as $N\to\infty$ only, since otherwise (17) holds (with $b_N = 1$) [1]–[3]. For the routing protocol to operate correctly all the required routing records need to be stored in nodes' memory, implying that b_N has to satisfy

$$b_N \geq \max\{1, \min_{n\in\mathcal{N}_N} \|T_N(n)\|\}$$

where $\mathcal{N}_N \subseteq \{1,\ldots,N\}$; recall that $b_N \geq 1$. The preceding relation and Theorem 2 yield a requirement on b_N

$$\mathbb{P}\left[b_N > \varepsilon/(\sqrt{a_N}\ln N)\right] \to 1 \tag{21}$$

for some $\varepsilon > 0$, as $N\to\infty$. Theorem 1 guarantees an existence of $\xi_3 < \infty$ such that

$$\liminf_{N\to\infty} \mathbb{P}\left[\vartheta_N > \frac{\xi_3}{\sqrt{a_N}N} \text{ is feasible}\right] < 1$$

when there is enough memory to store routing tables. Combining this inequality with (21) concludes the proof. Note that a certain throughput is feasible only if the buffer requirement is met. ■

V. Information-Efficient Routing

In this section, we establish that there exists an information-efficient routing algorithm. In particular, we show that column-first routing (see Section II) has such a property. That is, the scheme can operate correctly in large networks without sacrificing the throughput even when each node has enough memory to store only a finite number of routing records (addresses). The compression scheme described in Section III.C can be used to efficiently represent routing tables when column-first policy is employed. We stress that the column-first policy is only one of the possible information-efficient policies; we focus on this particular policy due to its simplicity. Also, it is feasible to devise an information-efficient policy that is distributed and does not require global node coordinates (see Section VI).

Consider a nonedge cell and a packet to be forwarded through this cell. If the packet is forwarded along a row, then the packet will remain in the same row until it reaches its destination. This implies that a packet arriving from the west or east neighboring cell has to be forwarded to the east and west neighbor, respectively. In other words, no packet is forwarded in the east-to-south, east-to-north, west-to-south and west-to-north direction ($\psi_d(i) = 0$ for these directions). For most of the cells, if a packet arrives from the south neighbor, it is more likely to be forwarded to the north neighbor, since

$$\mathbb{E}\psi_d(i) = a_N(l-1)(1 - l\sqrt{a_N})$$

where l is the row the cell is located in and d is the south-to-north direction, and

$$\mathbb{E}\left[\psi_{d_1}(i) + \psi_{d_2}(i)\right] = a_N(l-1)(\sqrt{a_N} - a_N)$$

where d_1 and d_2 are south-to-west and south-to-east directions, respectively.[1] Therefore, a cell can perform correct routing by maintaining only 4 sets of destination addresses $\{A_{d_i}\}$. Packets destined to addresses in $\{A_{d_i}\}$ are forwarded from the south or north cell to either the east or west cell, depending to which A_{d_i} the address belongs. All other packets are forwarded straight across the cell (north-to-south, south-to-north, east-to-west, west-to-east). With this kind of representation of the routing functions $R(\cdot,\cdot)$ each destination contributes at most one routing record to one cell in the network. This follows from the fact that each route experiences at most one 90°-change of direction.

The next proposition is the second main result of the paper.

Proposition 1: Let Π_b be the column-first routing algorithm in a network of nodes that are capable of storing b routing records. If $a_N > a\ln N/N$ for some $a > 1$, then there exists $b < \infty$ such that $\mathbb{P}[\Pi_b \text{ operates correctly}] \to 1$ as $N \to \infty$ under the cell-based algorithm.

Proof: Let $\nu_N(i)$ be the number of routes that undergo a change in direction in cell i when the network is of size N. Then, the maximum of $\nu_N(i)$ across all cells can be estimated using the union bound

$$\mathbb{P}\left[\max_{1\le i\le 1/a_N} \nu_N(i) \le \gamma a_N N\right] \ge 1 - \sum_{i=1}^{a_N^{-1}} \mathbb{P}[\nu_N(i) > \gamma a_N N].$$

For a fixed cell i, a route changes its direction at this cell only if the source is in the same column and the destination is in the same row (but not in the cell i). For $j = 1,\ldots,N$, let $I_j = 1$ if node j originates a flow that changes direction at cell i, and $I_j = 0$ otherwise. Then, we have

$$\nu_N(i) = \sum_{j=1}^{N} I_j. \tag{22}$$

Due to our assumption of independence of traffic routes (see Section II), the sequence $\{I_j\}_{j=1}^N$ is an i.i.d. sequence of Bernoulli random variables with $\mathbb{E}I_j = (\sqrt{a_N} - a_N)^2 < a_N$; the probability that the source node is in the same column as cell i (but not in cell i) is $(\sqrt{a_N} - a_N)$; the same applies for the destination. Then, (22), Markov's inequality and the independence of $\{I_j\}$ yield, for arbitrary $s > 0$

$$\begin{aligned}\mathbb{P}[\nu_N(i) > \gamma a_N N] &= \mathbb{P}\left[\prod_{j=1}^{N} e^{sI_j} > e^{s\gamma a_N N}\right]\\ &\le e^{-s\gamma a_N N}(\mathbb{E}e^{sI_1})^N\\ &\le e^{-s\gamma a_N N}(1 + a_N(e^s - 1))^N\\ &= e^{-s\gamma a_N N} e^{N\ln(1+a_N(e^s-1))}\\ &\le e^{-a_N N(\gamma s - e^s + 1)}\end{aligned}$$

where the last inequality is due to the fact that $\ln(1+x) \le x$ for $x \ge 0$. Given the assumptions of the proposition, it is possible to select s and γ such that, as $N \to \infty$

$$\mathbb{P}\left[\max_{1\le i\le 1/a_N} \nu_N(i) \le \gamma a_N N\right] \to 1. \tag{23}$$

The required amount of storage at individual nodes for the correct operation of the routing algorithm in cell i is upper bounded by $\|T_N(n)\| \le \zeta\lceil \nu_N(i)/k_N(i)\rceil$, where $\zeta > 1$ accounts for the number of bits needed to represent cell labels. This bound is based on the assumption that all records can be stored in a distributed fashion within the cell. Finally, (23) and Lemma 2 result in the equation shown at the bottom of the next page, as $N \to \infty$. Selecting $b = \lceil \zeta\gamma/\varepsilon\rceil$ completes the proof. ■

Theorem 1 from [10] and Proposition 1 suggest that nodes with limited memory can be used to create large multihop networks, at least when the transmission times are exponentially distributed. The protocol constructed in [10] utilized the column-first routing strategy. These two results indicate that memory bottlenecks can be avoided both in the data and control planes.

VI. Distributed Approach

In the previous sections, the discussion focused on a square-cell partition of the unit square. It is of interest to consider information-efficient routing algorithms with more general cell shapes. Moreover, column-first-like algorithms require high-level notions of rows and columns that do not

[1] The disparity of $\psi_d(i)$ for different directions that allows for an efficient representation of routing tables was exploited in [9] to obtain an approximation of the throughput of lattice networks with deterministic service times and finite buffers.

typically pre-exist in distributed networks. The purpose of this section is to illustrate that information efficiency of routing protocols is not tied to regular cell shapes and pre-existing notions of rows and columns; detailed design and analysis of efficient distributed algorithms is beyond the scope of this paper. Here, we provide an example on how virtual rows and columns can be distributively constructed with dense Voronoi tessellations. In the following discussion, cells in the network are not required to have any information about the network topology besides the identity of their neighboring cells.

Consider the Voronoi tessellation of the unit square (see [26] for a discussion on Voronoi tessellations). Let $\{x_1, x_2, \ldots, x_{1/a_N}\}$ be a set of $1/a_N$ points on the unit square; each point represents the center of a cell, and hence, there are $1/a_N$ cells in total. For each i, the Voronoi cell with center x_i is the set of all points that are closer to x_i than to any of the other x_j's, i.e., $\{x \in (0,1)^2 : |x - x_i| \leq |x - x_j|, j \neq i\}$. Such tessellation provides a method of grouping nodes that are physically close to each other and was considered in previous studies of cell-based forwarding algorithms. Following [1] (see Section IV in [1]), we require that each cell contains a disk of area $\Theta(a_N)$ as $N \to \infty$, and is contained by a disk of area $\Theta(a_N)$ as $N \to \infty$; this condition rules out cells that are eccentrically shaped. As in the previous sections, packets are forwarded between neighboring cells. The following procedure constructs virtual rows and columns for a Voronoi tessellated unit square (an instance is shown in Fig. 4):

1) A randomly selected cell c_0 floods a distance counter that increases with each hop over the network so that every cell is aware of the minimum distance (in hops) to c_0.
2) A cell c_1 with the longest hop distance to c_0 is identified. Cell c_1 performs another distance-counter flood to identify a furthest cell c_2 from c_1.
3) An initial path (a set of connected cells) P from c_1 to c_2 is created; random decisions are made whenever there is a tie.
4) Path P floods a distance counter; all of the cells in P have distance counters set to zero. The distance to P is equal to the minimum distance to a cell belonging to P. Cells with the same distance to P (and on the same side) form virtual rows.
5) A cell furthest away from P is identified as c_3; a cell furthest away from c_3 is identified as c_4. As in steps 3 and 4, cells c_3 and c_4 can be used to construct a path and thus the virtual columns.

Each time a distance counter is flooded, all cells receive the counter. Hence, all cells belong to exactly one virtual row and one virtual column. Note that neighboring cells in a virtual row (column) can be separated by at most one cell with a different distance counter, due to the nature of distance counters. Therefore, a cell needs to interact with its two-hop neighbors at most.

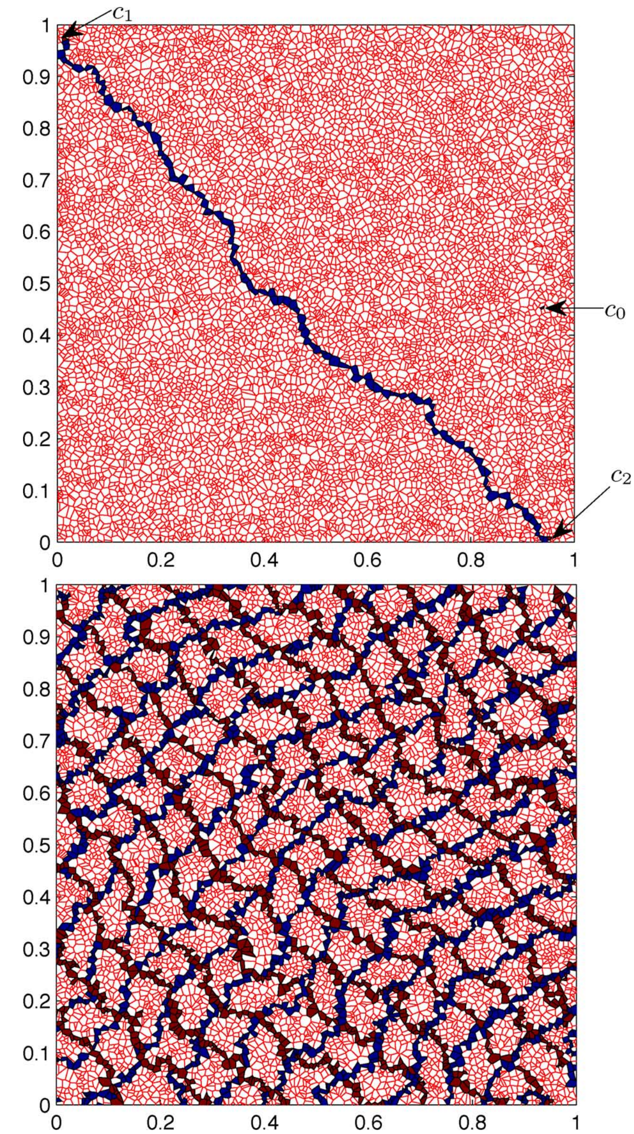

Fig. 4. Example on how virtual rows and columns for a Voronoi tessellated unit square are constructed by the distributed algorithm. 10000 points are selected uniformly at random as centers of Voronoi cells, which satisfy the condition that rules out too eccentric shapes. The figure on the top shows cells c_0, c_1, c_2, and the initial path P; in the figure on the bottom, only virtual rows and columns with indices of multiples of 6 are colored for clarity.

After a one-time construction of virtual rows and columns, a column-first-like routing algorithm can be performed on the unit square. Given a transmission pair, the source and destination nodes forward a request along the column and row they belong to. The two requests intersect at some cell, which in turn keeps the record of the forwarding destination address in the routing table. When a packet is forwarded along a virtual row or column, intermediate cells may need to forward the packet to multiple neighbors (i.e., the neighbors that have not yet received

$$\mathbb{P}\left[\max_{1\leq i\leq 1/a_N} \lceil \nu_N(i)/k_N(i) \rceil \leq \gamma/\varepsilon\right] \geq \mathbb{P}\left[\max_{1\leq i\leq 1/a_N} \nu_N(i) \leq \gamma a_N N, \min_{1\leq i\leq 1/a_N} k_N(i) \geq \varepsilon a_N N\right] \to 1$$

the packet). However, the number of forwarding actions taken remains constant. This is due to a constant number of neighbors for each cell. It is straightforward to argue that routing based on virtual rows and columns requires the same (order of magnitude) amount of routing information as column-first routing with square cells of the similar size. The above construction of virtual rows and columns can be generalized to a larger class of tessellations on convex regions. We stress that the above algorithm provides one way of constructing the virtual rows and columns, but is by no means exclusive.

VII. Proofs

This section contains proofs of Lemma 2 and Lemma 4.

A. Proof of Lemma 2

1) Statement (2): The union bound and the fact that all $\{k_N(i)\}_{i=1}^N$ are identically distributed yield

$$\mathbb{P}\left[\min_{1\le i\le 1/a_N} k_N(i)\ge\gamma_1 a_N N\right] \ge 1-a_N^{-1}\mathbb{P}[k_N(i)<\gamma_1 a_N N].$$

The last term in the preceding inequality can be bounded by applying Markov's inequality and invoking the independence of node locations on the unit square

$$\begin{aligned}\mathbb{P}[k_N(i)<\gamma_1 a_N N] &= \mathbb{P}\left[e^{-sk_N(i)} > e^{-s\gamma_1 a_N N}\right]\\ &\le e^{s\gamma_1 a_N N}\mathbb{E}e^{-sk_N(i)}\\ &\le e^{s\gamma_1 a_N N}[1-a_N(1-e^{-s})]^N\\ &= e^{s\gamma_1 a_N N + N\ln(1-a_N(1-e^{-s}))} \qquad (24)\end{aligned}$$

for some $s>0$. Combining (24) with the fact that $\ln(1-x)\le -x$, for all $x\in(0,1)$, results in

$$a_N^{-1}\mathbb{P}[k_N(i)<\gamma_1 a_N N] \le e^{-a_N N(1-e^{-s}-\gamma_1 s)-\ln a_N}.$$

Therefore, the lemma holds provided that the exponent on the right-hand side grows without bound as $N\to\infty$

$$a_N N(1-e^{-s}-\gamma_1 s)+\ln a_N \to \infty. \qquad (25)$$

For s small enough and γ_1 such that $1-e^{-s}-\gamma_1 s>0$, the function in (25) is monotonic in a_N, implying that it is lower bounded by

$$a(1-e^{-s}-\gamma_1 s)\ln N - \ln N + \ln\ln N + \ln a \qquad (26)$$

due to the assumption on a_N. Now, under the assumptions of the lemma, it is possible to select $s>0$ and $\gamma_1>0$ such that the function (26) tends to infinity as $N\to\infty$, and, hence, (25) holds. This concludes this part of the proof.

2) Statement (3): The proof is very similar to the one of the first part of the lemma. Namely, we have (27), shown at the bottom of the next page, for some $s>0$. With the assumptions of the lemma on a_N it is possible to select $\gamma_2<\infty$ such that the right-hand side of (27) tends to 0 as $N\to\infty$. Hence, the statement holds. ■

B. Proof of Lemma 4

1) Straight-Line Routing: In order to avoid repetition, we provide a detailed proof of the statement for only one specific direction d—south-to-east. All other values of d can be treated in a similar manner.

Let $\mathcal{C}_N$ be the set of cells located at least δ units away from the nearest edge of the unit square. The cardinality of such a set is at least $(1-\delta)a_N^{-1}$ due to the assumption $a_N\to 0$ as $N\to\infty$. Consider an arbitrary cell $i\in\mathcal{C}_N$.

First, we estimate the probability p that cell i forwards packets originating at an arbitrary node in the south-to-east direction. This can be achieved by conditioning on the location of the source node. To this end, suppose that the source node is located at distance r and angle φ relative to the upper-right corner of cell i as illustrated in Fig. 5. Denote this conditional probability by $p_{r,\varphi}$. The given cell forwards packets in the south-to-east direction if the straight line connecting the source and the respective destination crosses the lower and right edges of the cell (the bold edges in Fig. 5). For $\sqrt{2a_N}\le r\le\delta$ and $0\le\varphi\le\pi/4$, this occurs when the the destination is located in the shaded disc sector of angular size ϕ shown in Fig. 5. The fact that the radial size of the sector is δ and the definition of set $\mathcal{C}_N$ ensure that the sector is within the unit square for N large enough. The uniform distribution of destinations in the unit square yields that $p_{r,\varphi}$ is lower bounded by the area of the shaded sector, i.e.,

$$\begin{aligned}p_{r,\varphi} &\ge \phi[(\delta+r)^2-r^2]/2\\ &\ge \phi\delta^2/2. \qquad (28)\end{aligned}$$

Geometry of the considered case yields that the angle $\phi\le\varphi$ is related to the pair of polar coordinates (r,φ) via

$$\tan(\pi/2-\varphi-\phi)=\tan(\pi/2-\varphi)-\sqrt{a_N}/(r\sin\varphi)$$

which renders

$$r\sin\phi=\sqrt{a_N}\sin(\varphi-\phi) \qquad (29)$$

for $0<\varphi\le\pi/4$. By further restricting values of r and φ to $(\delta/2,\delta)$ and $(\pi/8,\pi/4)$, respectively, it is straightforward to obtain a lower bound on ϕ from (29)

$$\phi\ge\zeta\sqrt{a_N}/r \qquad (30)$$

for some $\zeta>0$ and all N large enough; the bound is based on the expansion of the $\sin$ function around the origin and $a_N\to 0$ as $N\to\infty$.

Now, integrating the conditional probability $p_{r,\varphi}$ over a restricted range of (r,ϕ) results in the following lower bound:

$$\begin{aligned}p &\ge \int_{2\sqrt{a_N}}^{\delta}\int_0^{\pi/4} p_{r,\varphi}\, r\, d\varphi\, dr\\ &\ge \int_{\delta/2}^{\delta}\int_{\pi/8}^{\pi/4}\frac{\phi\delta^2}{2}\, r\, d\varphi\, dr,\end{aligned}$$

where the second inequality is due to (28). The preceding relationship and (30) yield a desired bound for some $\xi>0$

$$p\ge\xi\sqrt{a_N}. \qquad (31)$$

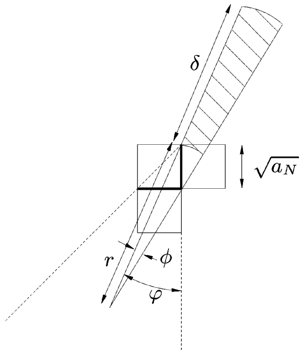

Fig. 5. Cell forwards packets in the south-to-east direction if the straight line connecting the source and destination crosses the two bold lines. Provided that the source is located at (r, φ) in polar coordinates relative to the cell's upper-right corner, the destination can be anywhere inside the shaded disc sector. The figure is plotted for the case $0 \le \varphi \le \pi/4$.

Second, consider an estimate of $\psi_d(i)$. The independence of node locations and (31) imply

$$\mathbb{P}[\psi_d(i) < \varepsilon\sqrt{a_N}N] \le \mathbb{P}\left[\sum_{j=1}^{N} I_j < \varepsilon\sqrt{a_N}N\right] \tag{32}$$

where $\{I_j\}_{j=1}^N$ is an i.i.d. sequence of Bernoulli random variables with $\mathbb{E}I_j = \xi\sqrt{a_N}$; variable I_j lower bounds the indicator function of whether node i forwards packets with the source node j in direction d.

Applying Markov's inequality to (32) results in, for $s > 0$

$$\begin{aligned}\mathbb{P}[\psi_d(i) < \varepsilon\sqrt{a_N}N] &\le e^{s\varepsilon\sqrt{a_N}N}[1 - \xi\sqrt{a_N}(1 - e^{-s})]^N \\ &\le e^{s\varepsilon\sqrt{a_N}N + N\ln(1-\xi\sqrt{a_N}(1-e^{-s}))} \\ &\le e^{-\sqrt{a_N}N(\xi(1-e^{-s})-\varepsilon s)}\end{aligned}$$

where the last inequality follows from $\ln(1-x) \le -x$ for all $x \ge 0$ small enough. For a given value of ξ, it is feasible to select s and $\varepsilon > 0$ such that $\xi(1-e^{-s}) - \varepsilon s > 0$. This fact, coupled with the assumption $\sqrt{a_N} \ge 1/\sqrt{N}$, yields the statement of the lemma.

2) Shortest-Path Routing: As in the proof of the statement for the straight-line scheme, we focus on one direction d—south-to-east; all other direction can be treated similarly.

Define $\mathcal{C}_N$ as the set of all cells that are at least $(\lceil\delta/\sqrt{a_N}\rceil + 1)$ *cells* away from the boundary of the unit square. The cardinality of such a set is at least $(1-\delta)a_N^{-1}$ for N large enough. Let i be an arbitrary cell in $\mathcal{C}_N$.

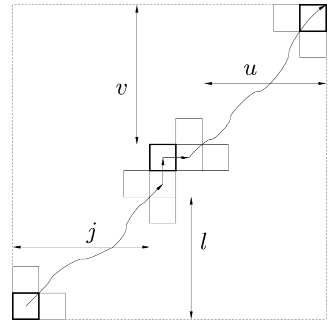

Fig. 6. Probability p of forwarding packets of a flow in the south-east direction by the cell in the center of the figure can be estimated by conditioning on the location of source and destination cells (lower-bottom and upper-left, respectively). All the distances shown are measured in cells.

Next, we lower bound the probability p that cell i forwards packets with the source at an arbitrary node in direction d. To this end, conditioning on the source and destination cells with relative locations (j, l) and (u, v), respectively (see Fig. 6), and a path-counting argument yield

$$\begin{aligned}p &\ge \frac{a_N^2}{4}\sum_{j,l=1}^{\lceil\delta/\sqrt{a_N}\rceil}\sum_{u,v=1}^{\lceil\delta/\sqrt{a_N}\rceil}\binom{j+l}{j}2^{-(j+l)} \\ &\ge \frac{a_N\delta^2}{4}\sum_{j=\lfloor\delta/(2\sqrt{a_N})\rfloor}^{\lceil\delta/\sqrt{a_N}\rceil}\sum_{l=\lfloor j-\sqrt{j}\rfloor}^{\lceil j\rceil}\binom{j+l}{j}2^{-(j+l)}.\end{aligned}$$

Since the relative values of j and l are large when N is large, Stirling's approximation is applicable and renders an estimate on the summands in the preceding inequality

$$\begin{aligned}\binom{j+l}{j}2^{-(j+l)} &\ge \frac{1}{2\pi}\sqrt{\frac{j+l}{jl}}\frac{(\frac{j+l}{2})^{j+l}}{j^j l^l} \\ &\ge \zeta/\sqrt{j}.\end{aligned}$$

for some $\zeta > 0$. Therefore, the following holds for N large enough

$$\begin{aligned}p &\ge \frac{a_N\delta^2}{4}\sum_{j=\lfloor\delta/(2\sqrt{a_N})\rfloor}^{\lceil\delta/\sqrt{a_N}\rceil}\sum_{l=\lfloor j-\sqrt{j}\rfloor}^{\lceil j\rceil}\zeta\left\lfloor\frac{\delta}{2\sqrt{a_N}}\right\rfloor^{-1/2} \\ &\ge \zeta\sqrt{a_N}\delta^3/16\end{aligned}$$

i.e., $p \ge \xi\sqrt{a_N}$, for some $\xi > 0$. The rest of the proof is identical to the second part of the proof for the straight-line scheme—see the part that follows (31). ■

$$\mathbb{P}\left[\max_{1\le i\le 1/a_N} k_N(i) \le \gamma_2 a_N N\right] \ge 1 - a_N^{-1}\mathbb{P}[k_N(i) > \gamma_2 a_N N]$$

and

$$\begin{aligned}a_N^{-1}\mathbb{P}[k_N(i) > \gamma_2 a_N N] &\le a_N^{-1}e^{-s\gamma_2 a_N N + N\ln(1+a_N(e^s-1))} \\ &\le e^{-a_N N(s\gamma_2+1-e^s)-\ln a_N}\end{aligned} \tag{27}$$

ACKNOWLEDGMENT

The authors would like to thank S. Savari for pointing out [20], [22]–[24]. The authors would also like to thank the anonymous reviewers for their comments.

REFERENCES

[1] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Feb. 2000.
[2] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, “Optimal throughput-delay scaling in wireless networks—Part I: The fluid model,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2568–2592, Jun. 2006.
[3] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, “Optimal throughput-delay scaling in wireless networks—Part II: Constant-size packets,” *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 5111–5116, Nov. 2006.
[4] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran, “On the throughput capacity of random wireless networks,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2756–2761, Jun. 2006.
[5] A. Jovičić, P. Viswanath, and S. Kulkarni, “Upper bounds to transport capacity of wireless networks,” *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2555–2565, Nov. 2004.
[6] O. Leveque and E. Telatar, “Information theoretic upper bounds on the capacity of ad hoc networks,” *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 858–865, Mar. 2005.
[7] P. R. Kumar and L.-L. Xie, “A network information theory for wireless communications: Scaling laws and optimal operation,” *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 748–767, May 2004.
[8] G. Barrenechea, B. Beferull-Lozano, and M. Vetterli, “Lattice sensor networks: Capacity limits, optimal routing and robustness to failures,” presented at the IPSN, Berkeley, CA, Apr. 2004.
[9] G. Barrenetxea, B. Beferull-Lozano, and M. Vetterli, “Efficient routing with small buffers in dense networks,” presented at the IPSN, Los Angeles, CA, Apr. 2005.
[10] P. Jelenković, P. Momčilović, and M. Squillante, “Scalability of wireless networks,” *IEEE/ACM Trans. Netw.*, vol. 15, no. 2, pp. 295–308, Feb. 2007.
[11] J. Herdtner and E. Chong, “Throughput-storage tradeoff in ad hoc networks,” presented at the IEEE Infocom, Miami, FL, Mar. 2005.
[12] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, “Geographic routing without location information,” presented at the ACM MobiCom, San Diego, CA, Sep. 2003.
[13] B. Karp and H. Kung, “Greedy perimeter stateless routing,” presented at the ACM MobiCom, Boston, MA, Aug. 2000.
[14] J. Li, J. Jannotti, D. D. Couto, D. Karger, and R. Morris, “A scalable location service for geographic ad hoc routing,” presented at the ACM MobiCom, Boston, MA, Aug. 2000.
[15] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, “Routing with guaranteed delivery in ad hoc wireless networks,” *Wireless Netw.*, vol. 7, pp. 609–616, 2001.
[16] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.
[17] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1992.
[18] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, and Hypercubes*. San Mateo, CA: Morgan Kaufmann, 1992.
[19] F. Kelly, *Reversibility and Stochastic Networks*. Hoboken, NJ: Wiley, 1979.
[20] G. Narlikar and F. Zane, “Performance modeling of fast IP lookups,” presented at the ACM Sigmetrics, Cambridge, MA, Jun. 2001.
[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ: Wiley, 1991.
[22] A. Buchsbaum, G. Fowler, and R. Giancarlo, “Improving table compression with combinatorial optimization,” presented at the ACM-SIAM SODA, San Francisco, CA, Jan. 2002.
[23] A. Orlitsky and J. Roche, “Coding for computation,” presented at the IEEE FOCS, Milwaukee, WI, Oct. 1995.
[24] E. Kushilevitz and N. Nisan, *Communication Complexity*. New York: Cambridge Univ. Press, 1997.
[25] S. Fuller, T. Li, J. Yu, and K. Varadhan, Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy RFC 1519,, Sep. 1993.
[26] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Hoboken, NJ: Wiley, 1992.

Li-Yen Chen received the B.S. degree in electrical engineering from the National Tsing-hua University, Taiwan, in 2005, and the M.S. and Ph.D. degrees in electrical engineering: systems from the University of Michigan, Ann Arbor, in 2007 and 2010, respectively.

Currently, she is a software development engineer.

Petar Momčilović received the Ph.D. degree from Columbia University, New York.

He is an Associate Professor in the Department of Industrial and Systems Engineering, University of Florida, Gainesville.