Lattice Networks: Capacity Limits, Optimal Routing and Queueing Behavior

Guillermo Barrenetxea, *Student Member, IEEE,* Baltasar Beferull-Lozano, *Member, IEEE,* and Martin Vetterli, *Fellow, IEEE*

Abstract

We study capacity and routing in lattice networks. Such networks are used in regular settings like grid computing and can be seen as an approximation to dense sensor networks. Thus, limits on capacity, optimal routing policies, and performance with finite queues are key issues and are addressed in this paper. In particular, we study the routing algorithms that achieve the maximum rate per node for infinite and finite buffers in the nodes and different communication models, namely uniform communications, central data gathering and border data gathering. In the case of nodes with infinite buffers, we determine the capacity of the network and we characterize the set of optimal routing algorithms that achieve capacity. In the case of nodes with finite buffers, we approximate the queue network problem and obtain the distribution on the queue size at the nodes. This distribution allows us to study the effect of routing into the queue distribution and derive the algorithms that achieve the maximum rate.

Index Terms

Lattice networks, square grid, torus, routing, queueing theory, network capacity, uniform communication, data gathering, border data gathering.

I. Introduction

Lattice networks are widely used, for example, in distributed parallel computation [2], distributed control [3] and wired circuits such as CMOS circuits and CCD-based devices [4]. Lattice networks are also known as grid [5] or mesh [6] networks. Moreover, the development of micro and nano-technologies [7] has also enabled the deployment of sensor networks for measuring and monitoring purposes [8]. The usual deployment of devices into the sensed area frequently consists of a regular structure that results into a lattice sensor network, or a perturbation of it.

Lattice networks can also be considered as an approximation for networks whose nodes are randomly located. Even though a particular realization of such random network does not correspond to a regular lattice structure, an average realization of this network can be usually approximated by a regular structure.

We consider lattice networks, namely the square lattice and torus lattice based networks. We choose these simple structures because they allow for a theoretical analysis while still being useful enough, as shown in this paper, to incorporate all the important elements, such as connectivity, scalability with respect to the size of the network and finite storage capacity.

In practice, common devices used in sensor networks have little storage (e.g. Berkeley motes have 512 KB [9]), and a similar lack of storage is typical in optical networks [10]. In this paper, we focus on the analysis and design of routing algorithms that maximize the throughput per node for networks with both infinite and finite buffer at the nodes. In the case of infinite buffers, we establish the fundamental limits of transmission capacity in lattice networks. We also characterize and provide optimal routing algorithms for which the rate per node is equal to the network capacity. These optimal routing algorithms satisfy the

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS, `http://www.mics.org`), a center supported by the Swiss National Science Foundation under grant number 5005-67322. The material in this paper was presented in part at Information Processing in Sensor Networks (IPSN), Berkeley, CA, April 2004 [1].

Guillermo Barrenetxea, Baltasar Beferull-Lozano and Martin Vetterli are with the School of Computer Science and Communication, Ecole Polytechnique Federale de Lausanne, Lausanne, CH-1015, Switzerland (guillermo.barrenetxea@epfl.ch, baltasar.beferull@epfl.ch, martin.vertterli@epfl.ch).

property of being space-invariant, i.e. the routing algorithm that is used to route the packets between any two nodes depends only on the relative position between them, not their absolute positions.

In the case of finite buffers, the analysis requires solving the queueing problem associated to the network. However, no analytical exact solutions are known for even the simplest queueing networks [11] and queueing approximations are used to model the network. We propose more accurate approximation models to the usual Jackson's Theorem that allow us to analyze and design new routing algorithms for finite buffer networks.

Depending on the structure and goal of the network (monitoring, data collection, actuation), nodes exhibit different communication patterns. In this work, we consider three different communication models that represent different network tasks: uniform communication, data gathering and border data gathering. In uniform communication, the probability of any node communicating to any other node in the network is the same for all pairs of nodes. It models a distributed control network where every node needs the information generated by all nodes in the network [12]. In data gathering, nodes only need to send their data to one common fixed node and corresponds to the case where one node (sink) collects the information generated by all the nodes in the network [13]. In border data gathering, the information generated by all nodes in the network is collected by the nodes located at the border of the square lattice. This network configuration models a situation that arises frequently in integrated devices. Nodes located on the borders can be easily connected to high-capacity transmission lines, while nodes inside the device are difficult to connect and can only communicate to neighbor nodes.

We assume that either the considered network is wired (e.g. a CMOS circuit) or if it is wireless, we assume contention is solved by the MAC layer. Thus, we abstract the wireless case as a graph with point-to-point links and transform the problem into a graph with nearest neighbor connectivity.

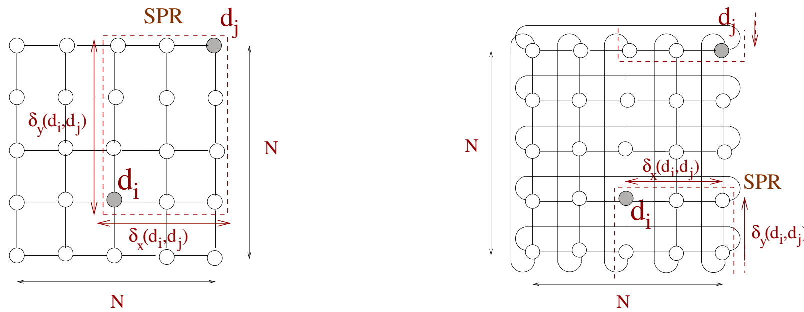
The rest of the paper is structured as follows. In Section II, we introduce the network model and our assumptions. In Section III, we study the uniform communication model under the infinite buffer assumption. We analyze capacity limits and provide optimal routing algorithms for both torus and square lattices. Then, in Section IV, we consider finite buffers and describe simple approximate models to analyze the corresponding queueing network. Using these models, we study the performance of routing algorithms under finite queues. In Section V, we carry out a similar analysis for the data gathering model and in Section VI we analyze the border data gathering problem. For both models, we characterize the optimal routing algorithms for both, infinite and finite buffers. Finally, conclusions are presented in Section VII.

A. Related Work

Regarding capacity, Gupta and Kumar studied the transport capacity in wireless networks [14] and concluded that, when specialized to lattices, the total end-to-end capacity per node is roughly $\mathcal{O}\left(1/\sqrt{n}\right)$ where n is the number of nodes.

Routing in lattice networks has been thoroughly studied in the context of distributed parallel computation [15], [2], where the system performance strongly depends on the routing algorithm. Various routing schemes have been studied through simulation by Maxemchuk [16].

Previous works that consider finite buffers are based on Jackson's theorem [11]. Harchol-Balter and Black [17] considered the problem of determining the distribution on the queue sizes induced by the greedy routing algorithm in torus and square lattice networks. They assumed that the time it takes for a packet to move through an edge is exponentially distributed. This hypothesis allows to reduce the problem into a product-form Jackson queue network and analyze it using standard queueing theory techniques. Although the exponential service time hypothesis is not realistic, they conjectured that it can be considered as an upper-bound for constant service time networks. This was confirmed by Mitzenmacher [18], who approximated the system by an equivalent Jackson network with constant service time queues. He provided bounds on the average delay and the average number of packets for square lattices for constant service times. Unfortunately, the separation of the upper and lower bound, in the general case, grows as the square root of the total number of nodes in the network. For an overview of packet routing in lattice networks, the reader is referred to [12].



(a) (b)

Fig. 1. Network model. (a) 5×5 square lattice and least displacement of $\{d_i, d_j\}$. (b) 5×5 wrapped square or torus lattice and least displacement of $\{d_i, d_j\}$. The shortest path region SPR(d_i, d_j) between nodes d_i and d_j is delimited in both cases by the dashed square.

Leighton [19] analyzed the performance of several routing algorithms for torus and square lattices. Based on probabilistic arguments, he provided bounds on the tail of the delay and queue size distributions.

Neely, Rohrs and Modiano's [20], [21] presented equivalent models for multi-stage tree networks of deterministic service time queues that reduces the analysis of tree network to the analysis of a much simpler two-stage equivalent model.

II. Model and Definitions

We consider graphs of size $N \times N$ nodes (or vertices) that are either a square or torus lattice. The subscripts "s" and "t" denote the square and the torus lattices respectively. The square lattice (Fig. 1(a)) is described by the graph $G_s(V, E_s)$ and the wrapped square or torus lattice (Fig. 1(b)) by the graph $G_t(V, E_t)$. A torus lattice network is obtained from a square lattice network by adding some supplementary links between opposite nodes located at the border of the lattice. Fig. 1 shows a $N \times N$ square and a torus lattices for $N = 5$.

Given a set S, let $|S|$ denote the cardinality of the set S. The $N \times N$ square lattice $G_s(V, E_s)$ contains $|V| = N^2$ nodes (or vertices) and $|E_s| = 2N(N-1)$ links (or arcs). The $N \times N$ torus lattice $G_t(V, E_t)$ contains $|V| = N^2$ nodes and $|E_t| = 2N^2$ links.

Every node in the network can potentially be the source or the destination of a communication, as well as a relay for communications between any other pair of nodes. We assume that nodes generate constant size packets and equal for all nodes following a stationary Bernoulli distribution with a constant average rate of $\mathcal{R}$ packets per time slot. We denote by $T(d_i, d_j)$ the probability of node d_i communicating to node d_j.

An arc or link $l \in E_{\{s,t\}}$ represents a communication channel between two nodes. In this work, we consider two cases for these communication channels, namely, the half-duplex and the full-duplex case, depending upon whether both nodes may simultaneously transmit, or whether one must wait for the other to finish before starting transmission. In the case of half-duplex links, if two neighbor nodes want to use the same link, we assume that both have the same probability of capturing the link for a transmission.

We denote the bandwidth of link l between d_i and d_j by $u(d_i, d_j)$. We assume that time is slotted and a one-hop transmission consumes one time slot[1], that is, $u(d_i, d_j) = 1$ for all $l \in E_{\{s,t\}}$. Moreover, we denote by $\varphi(d_i)$ the set of links connected to the node d_i.

The length of a path is defined as the number of links in that path. Moreover, we denote by $s(d_i, d_j)$ the length of the shortest path between nodes d_i and d_j. We define the *shortest path region* SPR(d_i, d_j) of a pair of nodes $\{d_i, d_j\}$ as the set of nodes that belong to any shortest path between d_i and d_j. For instance, SPR(d_i, d_j) in the square lattice is a rectangle with limiting corner vertices being d_i and d_j (Fig. 1(a)).

[1]For the sake of clarity we keep the subscripts d_i, d_j in our subsequent proofs.

For any pair of nodes $\{d_i, d_j\}$, we can view the lattice as an Euclidean plane map and consider d_j to be displaced from d_i along the X-Y Cartesian coordinates, where $\delta_x(d_i, d_j)$ and $\delta_y(d_i, d_j)$ are the relative displacements (Fig. 1(a)). We define the *least displacement* for these two nodes as $\delta(d_i, d_j) = [\delta_x(d_i, d_j), \delta_y(d_i, d_j)]$. Because of the particular existing symmetry in the torus lattice, given two nodes $\{d_i, d_j\}$, there are several possible values for $\delta(d_i, d_j)$. We consider $\delta(d_i, d_j)$ to be the one with the smallest norm (Fig. 1(b)).

We assume that nodes are equipped with buffer capabilities for the temporary storage of Q packets. When packets arrive at a particular node or are generated by the node itself, they are placed into a queue until the node has the opportunity to transmit them through the required link. Therefore, equivalently, we can consider that there are 4 queues per node, each one associated to one 4 output link.

Definition 1: The network capacity $C_{\{s,t\}}(N)$ is the maximum average number of information packets that can be transmitted reliably per node and per time slot in a network of size $N \times N$ with infinite buffer nodes.

A routing algorithm Π defines how traffic flows between any source destination pair $\{d_i, d_j\}$. Shortest path routing algorithms are those where packets transmitted between any two nodes d_i, d_j can only be routed inside SPR(d_i, d_j). We assume that routing algorithms are time invariant, that is, Π does not change over time. We further assume that nodes are not aware of their absolute positions in the network.

Definition 2: We say that a routing algorithm Π is *space invariant* if routing between any pair of nodes depend only on the relative position of the two corresponding nodes. That is, Π is space invariant if:

$$\text{If } \delta(d_i, d_j) = \delta(d_k, d_l) \text{ then } \Pi(d_i, d_j) = \Pi(d_k, d_l).$$

We denote by $\mathcal{R}^{\Pi}_{\text{max}}(N, Q)$ the maximum average rate that can be transmitted reliably per node and per time slot in a $N \times N$ lattice network with buffer size Q for a given routing algorithm Π. Obviously, $\mathcal{R}^{\Pi}_{\text{max}}(N, Q) \leq C(N)$.

In the next sections, we study network capacity and routing algorithms that achieve the maximum $\mathcal{R}^{\Pi}_{\text{max}}(N, Q)$ for different communication models.

III. Uniform Communication Model with Infinite Buffers

In the uniform communication model, the probability of any node communicating to any other node in the network is the same for all pairs of nodes, that is:

$$T(d_i, d_j) = \begin{cases} \frac{1}{N^2-1} & d_i \neq d_j, \\ 0 & d_i = d_j. \end{cases} \tag{1}$$

First, we study the network capacity and optimal routing with infinite buffers to obtain an absolute upper bound. Then, we analyze the effect of finite buffers in the network by proposing approximation models that allow us to simplify considerably the queueing network analysis.

A. Network Capacity

Under the infinite buffer hypothesis, the network capacity analysis is based only on stability issues. When the arrival rate is higher than the departure rate, queues become unstable and the expected delay is unbounded:

Lemma 1: The network capacity $C^u_{\{s,t\}}(N)$ for the uniform communication model is upper bounded as follows:

$$C^u_s(N) = \begin{cases} \frac{2\beta}{N}\left(1 - \frac{1}{N^2}\right), & \text{if } N \text{ is even,} \\ \frac{2\beta}{N}, & \text{if } N \text{ is odd,} \end{cases} \tag{2}$$

$$C^u_t(N) = \begin{cases} \frac{4\beta}{N}\left(1 - \frac{1}{N^2}\right), & \text{if } N \text{ is even,} \\ \frac{4\beta}{N}, & \text{if } N \text{ is odd,} \end{cases} \tag{3}$$

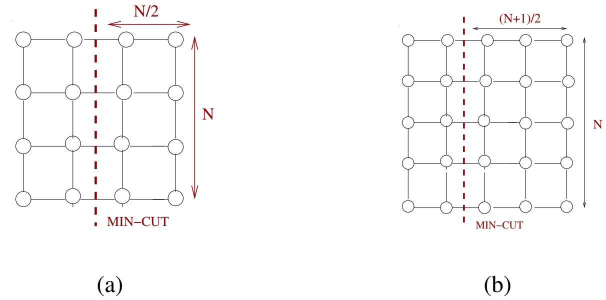


Fig. 2. Bisections for a $N \times N$ square lattice network. (a) N even. (b) N odd.

where β is equal to 1 for half-duplex links and 2 for full-duplex links.

Proof: This result follows directly from [19], where Leighton derived an upper bound for the square lattice based on bisection arguments. We can apply the same arguments to both torus and square lattices with the bisections shown in Fig. 2. Note that the capacity of the torus lattice is increased by a factor of 2 with respect to the square lattice network. This stems clearly from the fact that the number of links is increased while the traffic that flows across the bisection remains equal. ■

Note from (2) and (3) that, in both cases, the network capacity decreases with the square root of the total number of nodes, that is, with N. This decreasing behavior is also present in other kind of networks such as those presented in [14]. As we will see in next section, these upper bounds are actually tight and can be achieved under the infinite buffer assumption by certain routing algorithms.

B. Optimal Routing Algorithms

Network capacity (2,3) can indeed be achieved in both torus and square lattices by using the appropriate routing algorithms. In other words, $\max_{\Pi} \{\mathcal{R}^{\Pi}_{\text{max}}(N, \infty)\} = C^{u}_{\{s,t\}}(N)$.

Let $F^{\Pi}(d_i, d_j, d_k)$ be the traffic generated at node d_i with destination node d_j that flows through node d_k according to a particular routing algorithm Π. Similarly, we denote by $\lambda^{\Pi}_{d_k}$ the traffic arrival rate to node d_k according to a routing algorithm Π. Therefore:

$$\lambda^{\Pi}_{d_k} = \sum_{d_i \in V} \sum_{d_j \in V}^{N^2} T(d_i, d_j) F^{\Pi}(d_i, d_j, d_k). \tag{4}$$

The next proposition characterizes the class of shortest path routing algorithms that are optimal for torus lattice networks.

Theorem 1: A shortest path routing algorithm Π achieves network capacity for the torus lattice network if Π is space invariant. That is:

$$\text{if } \Pi \in \{\text{shortest path space invariant}\} \text{ then } \mathcal{R}^{\Pi}_{\text{max}}(N, \infty) = C^{u}_{t}(N).$$

Proof: Given the structural periodicity of the torus, if Π is space invariant, for every source-destination pair $\{d_{i1}, d_{j1}\}$ that generates traffic flowing across any particular node d_{k1}, there always exists another source-destination pair $\{d_{i2}, d_{j2}\}$ with the same least displacement as $\{d_{i1}, d_{j1}\}$ that generates exactly the same traffic flowing across some other node d_{k2} in the network (Fig. 3). That is:

$$\forall\, d_{k2} \in V,\ \exists \{d_{i2}, d_{j2}\} : \{\delta(d_{i2}, d_{j2}) = \delta(d_{i1}, d_{j1}) \text{ and } F^{\Pi}(d_{i2}, d_{j2}, d_{k2}) = F^{\Pi}(d_{i1}, d_{j1}, d_{k1})\}.$$

Consequently, the arrival rate to any node in the network is constant. That is,

$$\text{if } \Pi \in \{\text{ space invariant }\}, \quad \lambda^{\Pi}_{d_k} = \lambda \quad \text{for all } d_k \in V. \tag{5}$$

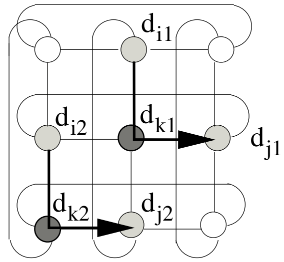


Fig. 3. The source-destination pair $\{d_{i_1}, d_{j_1}\}$ generates traffic that flows across node d_{k_1} according to Π. If Π is space invariant, for any other node d_{k_2}, we can find another source-destination pair $\{d_{i_2}, d_{j_2}\}$ with the same least displacement as $\{d_{i_1}, d_{j_1}\}$ that generates exactly the same traffic across d_{k_2} as $\{d_{i_1}, d_{j_1}\}$ across d_{k_1} .

Let $\overline{L}(N)$ be the average distance between a source and a destination for a given communication model described by $T(d_i, d_j)$. Then,

$$\overline{L}(N) = \frac{1}{N^2} \sum_{d_i \in V} \sum_{d_j \in V} T(d_i, d_j) s(d_i, d_j). \tag{6}$$

Particularly, for the uniform communication model, the average distance is given by:

$$\overline{L}(N) = \frac{1}{N^2 - 1} \sum_{d_j \in V} \sum_{d_j \in V \backslash d_i} s(d_i, d_j). \tag{7}$$

In the uniform communication model, all nodes generate packets at a constant rate $\mathcal{R}$. These packets take, on average, $\overline{L}(N)$ hops before reaching their destination. Therefore, the total traffic per unit of time generated in the network is given by $N^2 \mathcal{R} \overline{L}(N)$. If Π is space invariant, according to (5), all nodes have the same average rate and the total traffic is therefore uniformly distributed among all nodes. That is, the arrival rate λ at any node is given by:

$$\lambda = \frac{N^2 \mathcal{R} \overline{L}(N)}{N^2} = \mathcal{R} \overline{L}(N). \tag{8}$$

The average distance between any source node and any destination node in a $N \times N$ torus lattice under uniform traffic distribution is given by [22]:

$$\overline{L}(N) = \begin{cases} \frac{N^3}{2(N^2-1)} & \text{if } N \text{ is even} \\ \frac{1}{2} N & \text{if } N \text{ is odd.} \end{cases} \tag{9}$$

The stability condition in the nodes is given by:

$$\rho = \frac{\lambda}{\mu} < 1, \tag{10}$$

where μ is the average number of packets transmitted per unit of time. In the limit, as $\rho \to 1$, $\mu = 4$ for full-duplex communication channels and $\mu = 2$ for half-duplex.

Therefore, combining (8) and (10), the maximum rate per node $\mathcal{R}^{\Pi}_{\text{max}}(N)$ achieved under any space invariant routing algorithm Π is given by:

$$\mathcal{R}^{\Pi}_{\text{max}}(N) = \frac{2\beta}{\overline{L}(N)}. \tag{11}$$

Combining (9) and (11), $\mathcal{R}^{\Pi}_{\text{max}}(N)$ is equal to the upper bound given in (3). ■

As a consequence of Theorem 1, we have the following achievability result:

Corollary 1: The network capacity $C^u_t(N)$ of a torus lattice network is equal to the upper bound given by (3).

Theorem 1 says that, given the structural periodicity of the torus, the use of space invariant routing algorithms induces a uniform traffic distribution in the network that guarantees the maximum rate per node.

This uniform traffic distribution is not possible in the case of a square lattice network. Given the topology of a square lattice, as a node is located closer to the geographic center of the lattice, it belongs to the SPR of an increasing number of source-destination pairs. In the case of shortest path routing algorithms, this implies a higher traffic load. For the sake of simplicity, we restrict our analysis to the case of odd N. The analysis for even N is similar but more cumbersome, while essentially the same results hold. Notice also that since we are interested in large networks (large N), this is not a limiting restriction.

Theorem 2: For the square lattice network and the uniform communication model, the total average traffic $\lambda_{d_c}^{\Pi}$ that flows through the center node d_c for any space invariant routing algorithm Π, is lower bounded by:

$$\lambda_{d_c}^{\Pi} \geq \mathcal{R}N. \tag{12}$$

Proof:

The prove is constructive: we show that this lower bound is actually tight and design a routing algorithm Π that achieves this lower bound. In this proof we make use of the concept of least displacement and the property of space invariant routing algorithms.

Let $\Gamma^{\Pi}(\delta_x, \delta_y, d_c)$ denote the traffic generated by all pair of nodes with a least displacement given by $[\delta_x, \delta_y]$ that flows through node d_c, that is,

$$\Gamma^{\Pi}(\delta_x, \delta_y, d_c) = \sum_{d_i \in V} \sum_{\substack{d_j \in V: \\ \delta(d_i, d_j) = [\delta_x, \delta_y]}} T(d_i, d_j) F^{\Pi}(d_i, d_j, d_c).$$

Given the symmetry of d_c, $\Gamma^{\Pi}(\delta_x, \delta_y, d_c)$ has the following properties:

$$\Gamma^{\Pi}(\delta_x, \delta_y, d_c) = \Gamma^{\Pi}(\delta_y, \delta_x, d_c). \tag{13}$$

$$\Gamma^{\Pi}(\delta_x, \delta_y, d_c) = \Gamma^{\Pi}(|\delta_y|, |\delta_x|, d_c). \tag{14}$$

The traffic arrival rate to d_c can be obtained by summing over all possible least displacements in the network:

$$\lambda_{d_c}^{\Pi} = \sum_{\delta_x = -(N-1)}^{N-1} \sum_{\delta_y = -(N-1)}^{N-1} \Gamma^{\Pi}(\delta_x, \delta_y, d_c). \tag{15}$$

Using properties (13) and (14), we reduce the analysis of $\Gamma^{\Pi}(\delta_x, \delta_x, d_c)$ in (15) to the case $\delta_x \geq \delta_y$. That is,

$$\lambda_{d_c}^{\Pi} = 4 \sum_{\delta_x=0}^{N-1} \Gamma^{\Pi}(\delta_x, \delta_x, d_c) + 4 \sum_{\delta_x=0}^{N-1} \Gamma^{\Pi}(\delta_x, 0, d_c) + 8 \sum_{\delta_x=2}^{N-1} \sum_{\delta_y=1}^{\delta_x - 1} \Gamma^{\Pi}(\delta_x, \delta_y, d_c). \tag{16}$$

To derive now an lower bound for $\lambda_{d_c}^{\Pi}$, we can equivalently compute an lower bound for $\Gamma^{\Pi}(\delta_x, \delta_y, d_c)$ and apply (16). To compute $\Gamma^{\Pi}(\delta_x, \delta_y, d_c)$ we add the traffic contribution $F^{\Pi}(d_i, d_j, d_c)$ of all source-destination pairs $\{d_i, d_j\}$ such that $\delta(d_i, d_j) = [\delta_x, \delta_y]$. Instead of keeping d_c fixed and compute the traffic that goes through d_c for all $\{d_i, d_j\}$ such that $\delta(d_i, d_j) = [\delta_x, \delta_y]$, we can equivalently consider a fixed rectangle $R_c(\delta_x, \delta_y)$ of size $[\delta_x, \delta_y]$ and locate d_c in several positions. In other words, we determine the set $\mathcal{S}$ of relative positions of d_c in $R_c(\delta_x, \delta_y)$ with respect to all source destination pairs $\{d_i, d_j\}$ such that $\delta(d_i, d_j) = (\delta_x, \delta_y)$. Then, the traffic that flows through d_c for any shortest path routing algorithm Π can be computed as the total traffic generated by Π in $\mathcal{S}$. Fig. 4 shows an example for $[\delta_x, \delta_y] = [3, 2]$.

Once we obtain $\mathcal{S}$, we construct the routing policy Π that minimizes the total average traffic flowing through the set $\mathcal{S}$ or, equivalently, that minimizes $\Gamma^{\Pi}(\delta_x, \delta_x, d_c)$.

First note that if $\delta_y \geq \frac{N-1}{2}$, the set $\mathcal{S}$ has a vertical size smaller than δ_y and consequently, $\mathcal{S}$ does not fill completely any column of $R_c(\delta_x, \delta_y)$. We can therefore design a routing policy that uses only nodes in

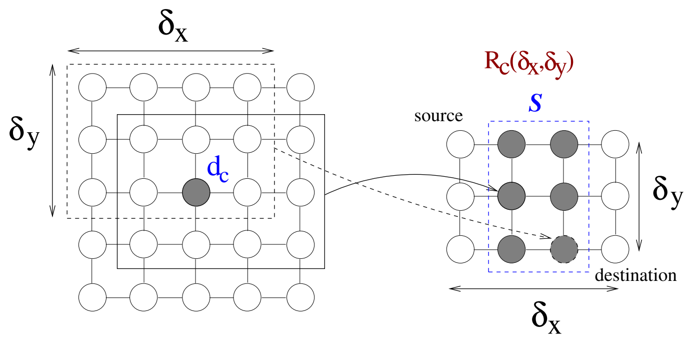


Fig. 4. For all source-destination pairs $\{d_i, d_j\}$ such that $\delta(d_i, d_j) = [3, 2]$, we obtain the set $\mathcal{S}$ of relative positions of d_c in $R_c(\delta_x, \delta_y)$

the set $R_c(\delta_x, \delta_y) \setminus \mathcal{S}$ and that generates no traffic in $\mathcal{S}$. The only routing policy that fulfills this condition for all $\delta_y \geq \frac{N-1}{2}$ consists on using only the two most external paths of SPR(d_i, d_j) (Fig. 5(a)). Fig. 5(a) illustrates this case in a 5×5 square lattice where $\delta(d_i, d_j) = [4, 3]$. Therefore, in general, for any routing policy Π,

$$\text{If } \delta_y > \frac{N-1}{2}, \quad \Gamma^{\Pi}(\delta_x, \delta_y, d_c) \geq 0, \text{ for all } \delta_x.$$

If $\delta_y > \frac{N-1}{2}$ we distinguish between two cases. If $\delta_x > (N-1)/2$, $\mathcal{S}$ fills completely $N - \delta_x$ columns of $R_c(\delta_x, \delta_y)$. Therefore, all routes between the source and the destination go through at least one node belonging to each of these $N - \delta_x$ columns. Given that $T(d_i, d_j) = 1/(N^2 - 1)$ for all $d_i, d_j \in V,\ d_j \neq d_i$, the total traffic that goes through $\mathcal{S}$ is lower bounded by:

$$\text{If } \delta_x > \frac{N-1}{2}, \delta_y < \frac{N-1}{2}, \quad \Gamma^{\Pi}(\delta_x, \delta_y, d_c) \geq \frac{\mathcal{R}}{N^2-1}\left(N - \delta_x\right).$$

Fig. 5(b) illustrates this case in a 5×5 square lattice where $\delta(d_i, d_j) = [3, 2]$. In this case, there are many routing policies that achieve this lower bound. For instance, a routing algorithm that uses only the two most external paths achieves the lower bound.

If $\delta_y \leq \frac{N-1}{2}$, $\mathcal{S}$ fills all the δ_x columns of $R_c(\delta_x, \delta_y)$ and any route between the source and the destination crosses at least $\delta_x + \delta_y$ nodes belonging to $\mathcal{S}$. Note that we only consider the locations of d_c as a source of a transmission and not as a destination. Obviously, the packets that reach d_c and have d_c as final destination do not interfere with the traffic going through d_c, while the traffic generated at d_c itself does. Fig. 5(c) illustrates this case in a 5×5 square lattice where $\delta(i, j) = (2, 2)$. Therefore:

$$\text{If } \delta_x \leq \frac{N-1}{2}, \delta_y < \frac{N-1}{2}, \quad \Gamma^{\Pi}(\delta_x, \delta_y, d_c) = \frac{\mathcal{R}}{N^2-1}\left(\delta_x + \delta_y\right)).$$

Putting all three cases together, we have that $\Gamma^{\Pi}(\delta_x, \delta_y, d_c)$ is lower bounded as follows:

$$\Gamma^{\Pi}(\delta_x, \delta_y, d_c) \geq \begin{cases} \frac{\mathcal{R}}{N^2-1}(\delta_x + \delta_y) & \delta_x \leq \frac{N-1}{2}, \delta_y \leq \frac{N-1}{2}, \\ \frac{\mathcal{R}}{N^2-1}(N - \delta_x) & \delta_x > \frac{N-1}{2}, \delta_y \leq \frac{N-1}{2}, \\ 0 & \text{otherwise}, \end{cases} \tag{17}$$

and routing algorithm Π that achieves minimization in the three cases consists in flowing data only through the most external paths.

Using (17) into (16), we bound the total traffic that flows through d_c as:

$$\lambda_{d_c}^{\Pi} \geq \left(\frac{\mathcal{R}}{N^2-1}\right)\left\{4\sum_{\delta_x=0}^{(N-1)/2} 2\delta_x + 4\sum_{\delta_x=0}^{(N-1)/2} \delta_x + 4\sum_{\delta_x=(N+1)/2}^{N-1} (N - \delta_x)\right.$$
$$\left. + 8\sum_{\delta_x=2}^{(N-1)/2}\sum_{\delta_y=1}^{\delta_x-1}(\delta_x + \delta_y) + 8\sum_{\delta_x=(N+1)/2}^{N-1}\sum_{\delta_y=1}^{(N-1)/2}(N - \delta_x)\right\}.$$

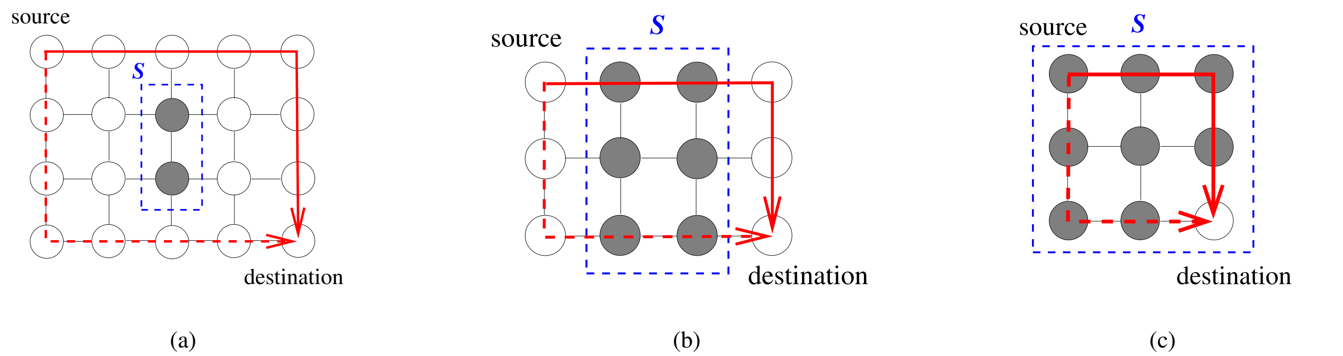


Fig. 5. $R_c(\delta_x, \delta_x)$ for three possible cases in a 5×5 lattice network: (a) $\delta(i,j) = (4,3)$; since $\delta_y \geq \frac{N-1}{2}$, the set of relative positions of d_c does not fill completely any column of SPR(d_i, d_j). (b) $\delta(i,j) = (3,2)$; since $\delta_y \leq \frac{N-1}{2}$, the set of relative positions of d_c fills $N - \delta_x$ columns completely. (c) $\delta(i,j) = (2,2)$; The set of relative positions fills all the δ_x columns and any route between the source and the destination has to cross $\delta_x + \delta_y$ relative positions. The arrows indicates two of the possible routing policies that generates the least possible traffic in $\mathcal{S}$.

and evaluating summations yields to:

$$\lambda_{d_c}^{\Pi} \geq \mathcal{R}N.$$

■

As a consequence of Theorem 2, we have the following corollaries:

Corollary 2: A shortest path space invariant routing algorithm achieves capacity in the uniform communication model only if the total average traffic $\lambda_{d_c}^{\Pi}$ that flows through the center node d_c is greater or equal to the total average traffic flowing through any other node d_x, that is:

$$\lambda_{d_c}^{\Pi} \geq \lambda_{d_x}^{\Pi}, \quad \text{for all } d_x \in V/d_c.$$

Proof: The proof follows by contradiction. Suppose a network capacity achieving routing algorithm Π that generates a traffic distribution where there exists a node d_x such that $\lambda_{d_x}^{\Pi} > \lambda_{d_c}^{\Pi}$. By Theorem 2,

$$\lambda_{d_x}^{\Pi} > \mathcal{R}N.$$

Imposing stability conditions for d_x, $\rho = \frac{\lambda_{d_x}}{\mu} < 1$, in both the full-duplex and the half-duplex case, we obtain that the maximum rate $\mathcal{R}_{\text{max}}^{\Pi}(N, \infty)$ achieved by Π is:

$$\mathcal{R}_{\text{max}}^{\Pi}(N, \infty) < C_s^u(N),$$

and therefore, Π does not reach capacity. ■

Corollary 2 says that the factor that really limits the maximum achievable rate in the network is the amount of traffic routed through the center node d_c. Intuitively, to maximize the maximum achievable rate per node $\mathcal{R}_{\text{max}}^{\Pi}(N, \infty)$, a routing algorithm has to avoid routing packets through the lattice center and promote as much as possible the distribution of traffic towards the borders of the lattice. In this way, we compensate the higher number of paths passing through the center of the lattice by enforcing a lower average traffic for these paths.

In the proof of Theorem 2 we have already characterized the set of routing policies that generates the minimum traffic in d_c. One of these policies consists on flowing data only along the two most external paths of the source and the destination SPR. In other words, nodes always route packets along the row (or column) in which they are located towards the destination node until they reach the destination's column (or row). Then, packets are sent along the destination's column (row) until they reach the destination node (see Fig. 6). We denote this routing by *row-first* (*column-first*)[19]. Note that both algorithms (row-first and column-first) are equivalent because of the lattice symmetry.

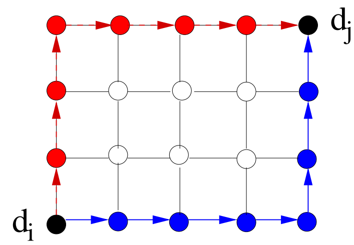


Fig. 6. Row-first (solid lines) and column-first (dashed lines) routing algorithm. Nodes route packets using the most external paths.

Corollary 3: For the square lattice network and the uniform communication model, the maximum average rate per node $\mathcal{R}^{\text{r-f}}_{\max}(N,\infty)$ achieved by the row-first routing algorithm is the maximum possible rate, that is, $\mathcal{R}^{\text{r-f}}_{\max}(N,\infty) = C^u_s(N)$.

Proof: From Theorem 2, we know that:

$$\lambda^{\text{r-f}}_{d_c} = \mathcal{R}N.$$

It is easy to verify that for the row-first routing, the network traffic distribution is such that:

$$\lambda^{\text{r-f}}_{d_c} > \lambda^{\text{r-f}}_{d_x} \quad \forall\, d_x \in V/d_c.$$

Imposing stability conditions for d_c, $\rho = \frac{\lambda^{\text{r-f}}_{d_c}}{\mu} < 1$, in both the full-duplex and the half-duplex case, we obtain that the maximum rate $\mathcal{R}^{\text{r-f}}_{\max}(N,\infty)$ achieved by row-first routing is:

$$\lambda^{\text{r-f}}_{\max}(N,\infty) = C^u_s(N).$$

■

As a consequence, we have the following achievability result:

Corollary 4: The network capacity $C^u_s(N)$ of a square lattice network is actually equal to the upper bound given by (2).

IV. Uniform Communication with Finite Buffers

Notice that with finite buffers, the maximum rate per node is clearly reduced due to buffer overflow. Overflow losses will first appear in the most loaded node or nodes, and consequently, these are the nodes that determine the maximum achievable rate $\mathcal{R}^{\Pi}_{\max}(N,\infty)$.

In a square lattice, the node located in the center of the network is clearly the most loaded node (Lemma 2). In a torus, if the routing algorithm is space invariant, all nodes support exactly the same traffic on average (Theorem 1) and, furthermore, all nodes in the network are indistinguishable due to the torus symmetry. Therefore, we can consider that the most loaded node is any node in the network. For both torus and square lattices, we denote the most loaded node as d_m.

We can restrict our analysis to the routing algorithm that achieves capacity with infinite buffers in both torus and square lattices, namely, row-first routing. Moreover, we show latter in this section that row-first routing is also optimal for finite queues.

Computing the network capacity for different buffer sizes Q requires analyzing a queueing network and computing the distribution on the queue size at d_m. However, the analysis of queueing networks is complex and usually no analytical exact solutions are known [11]. In the following, we introduce some approximations that simplify this analysis. First, we approximate the network by a tree of deterministic service time queues that can be reduced to a simple two-stage network. The analysis of this approximated two-stage network provides meaningful theoretical results that, as shown later, are close to the results obtained by simulation

We can decompose d_m into four identically distributed and independent FIFO queues associated to its four output links. The input packets to d_m whose final destination is not d_m are sent through one of

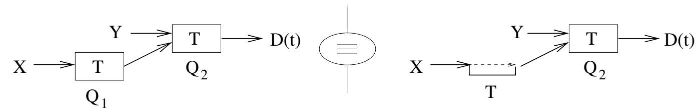


Fig. 7. The total number of packets in a two-queue system remains the same if the first stage queue is replaced by a pure delay of T time slots.

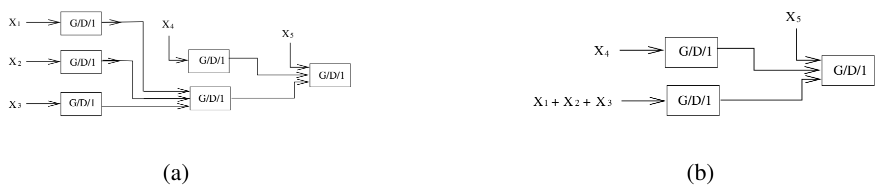


Fig. 8. The number of packets in the head node of the tree network (a) is the same as in the two-stage equivalent model (b).

the four output links depending on their destinations. In view of the symmetry of d_m for both torus and square lattices, the arrival distributions to these four links are equal. Moreover, due to the independence of packet generation, we assume that these arrival distributions are also independent. Then, we approximate the distribution on the queue size at d_m as the addition of these four iid distributions and compute it as the convolution of each individual queue. Therefore, we reduce the problem to computing the distribution on the size of only one of these iid queues q_m at d_m associated to one of the output links l_m.

Next, we propose different approximations for full-duplex and half-duplex links (whether l_m is half-duplex or full-duplex) and compare them with experimental results.

1) Full-Duplex communication channels: For full-duplex channels, q_m has a dedicated link and, since all packets have the same size, it can be modeled as a deterministic service time queue.

In the approximation model, we use some results by Neely, Rohrs and Modiano's [20], [21] on equivalent models for multi-stage tree networks of deterministic service time queues. We begin by reviewing the main results in [20], [21], and then we show how these results can be applied to our problem.

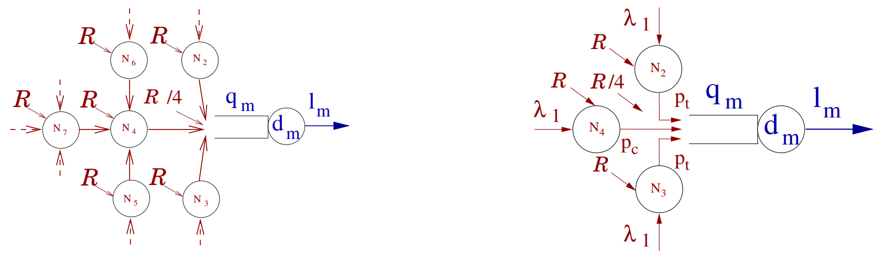
Theorem 3: ([20]) The total number of packets in a two-queue system is the same as in a system where the first stage queue has been replaced by a pure delay of T time slots.

Theorem 4: ([21]) The analysis of the queue distribution in the head node of a multi-stage tree system can be reduced to the analysis of a much simpler two-stage equivalent model, which is formed by considering only nodes located one stage away from the head node and preserving the exogenous inputs.

Fig. 7 shows the equivalence provided by Theorem 3. Fig. 8(a) shows a tree system and Fig. 8(b) its two-stage equivalent model. Importantly, these equivalences do not require any assumption about the nature of the input traffic. The only necessary condition is that all queues of the tree network have a deterministic service time T, and the input traffic is stationary and independent among sources.

We use these results to obtain the distribution on the size of q_m. First, we identify q_m as the head node of a tree network composed by all nodes sending traffic through l_m (Fig. 9(a)). Applying Theorem 4, the distribution on the size of q_m can be approximated by the distribution at the head node of the two-stage model (Fig. 9(b)), where we only consider the three neighbors located one hop away from d_m and preserve the traffic generated by the entire network that flows through l_m.

Note that the tree network associated to q_m (Fig. 9(a)) does not correspond exactly to the tree network of Theorem 4 (Fig. 8(a)). The reason is that, in addition to the exogenous inputs generated at each node, we also have some traffic leaving the network corresponding to the traffic that reached its destination. Under the uniform communication model, the average traffic leaving the network at any node is equal to $\mathcal{R}$. However, as the network size increases, the departing traffic at each node decreases as $\mathcal{O}(1/N)$ and consequently it is low compared to the traffic that flows through them. Hence, the two-stage model provides an approximated network.



(a) (b)

Fig. 9. Tree network and equivalent two-stage model. (a) Tree network associated to one output link l_m of d_m. (b) Two-stage model of the tree network.

The arrivals to the nodes of the first stage in the two-stage model correspond to the addition of all exogenous inputs routed through l_m. Since packets are generated in sources following independent Bernoulli distributions, this arrival process converges, as the number of nodes increases, to a Poisson distribution.

In the uniform communication model, packets travel $\mathcal{O}(N)$ hops on average before reaching their destination. Using row-first routing, packets travel most of the time along the same row or column, turning only once. Consequently, the traffic entering a node by a row or a column link continues with high probability along the same row or column. Let p_c denote the probability of a packet to continue along the same row or column, and p_t the probability of turning. These probabilities are easy to calculate for d_m:

$$p_c = \begin{cases} \frac{N-1}{N+1}, & \text{for the square lattice,} \\ \frac{N/2-1}{N/2+1}, & \text{for the torus.} \end{cases} \tag{18}$$

$$p_t = \begin{cases} \frac{N-1}{2N(N+1)}, & \text{for the square lattice,} \\ \frac{N/2-1}{N(N/2+1)}, & \text{for the torus.} \end{cases} \tag{19}$$

Note that p_c goes asymptotically to one as the number of nodes increases while p_t goes to zero. It follows that q_m receives most of the traffic from the node located in the same row or column.

Apart from the traffic that arrives from its neighbors, d_m generates also new traffic that is injected to the network at a rate $\mathcal{R}$. Considering again the symmetry of d_m, the fraction of this traffic that goes through l_m is $\mathcal{R}/4$. The average arrival rate λ_{q_m} to q_m can be computed as the addition of the traffic generated in d_m and the traffic arriving from its neighbors:

$$\lambda_{q_m} = \mathcal{R}/4 + \lambda_1 \left(p_c + 2p_t\right), \tag{20}$$

where λ_1 is the total arrival rate to the neighbors of d_m (Fig. 9(b)).

For row-first routing and the uniform communication model, λ_1 is equal to:

$$\lambda_1 = \begin{cases} \frac{\mathcal{R}N}{4}, & \text{for the square lattice,} \\ \frac{\mathcal{R}N}{8}, & \text{for the torus.} \end{cases} \tag{21}$$

We can express $\mathcal{R}$ as a fraction of the network capacity $C(N)$, that is, $\mathcal{R} = \alpha C(N)$, and denote α as relative capacity. Then, $\lambda_1 = \alpha$ for both torus and square lattices. Putting everything together, the resulting approximation model is shown in Fig. 11(a).

Regardless of the number of nodes in the network, we reduce the analysis of the distribution on the size of q_m to a four queue network. This approximation holds for any input traffic distribution as long as it is stationary and independent among the different sources.

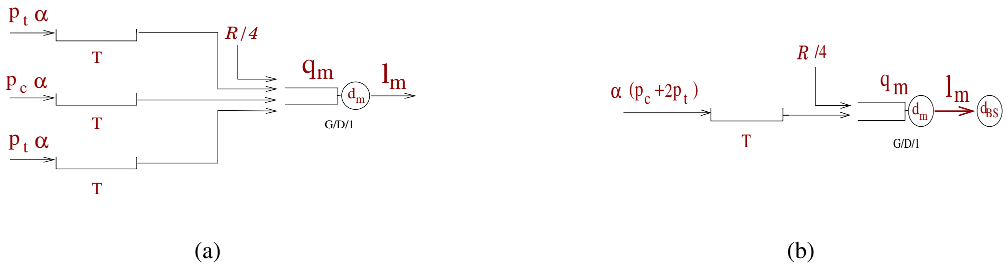


Fig. 10. Two-stage equivalent networks. (a) If we replace the queues of the first stage by pure delays of T time slots, the total number of packets in the approximated model remains constant. (b) In terms of number of packets, this is equivalent to injecting all the arrivals to a single pure delay.

Theorem 5: The buffer size required to achieve a certain relative capacity α decreases with the network size N. Furthermore, the required buffer size goes asymptotically to zero.

Proof: By Theorem 3, the total number of packets in the approximated model (Fig. 11(a)) is the same as a system where the queues of the first stage have been replaced by pure delays of T time slots (Fig. 10(a)). We can decompose the total number of packets $S(t)$ in the system as the number of packets in the first stage $S_1(t)$ plus the number of packets in the head node $S_h(t)$, that is, $S(t) = S_1(t) + S_h(t)$.

The total average arrival rate λ_{S_1} to the first stage is

$$\lambda_{S_1} = \alpha(p_c + 2p_t) = \alpha(N-1)/(N+1). \tag{22}$$

In terms of number of packets this is equivalent to injecting all the arrivals to a single pure delay (Fig. 10(b)). Consequently, for a fixed α, since λ_{S_1} is almost constant for large N, $S(t)$ is also almost constant.

Moreover, as N increases, p_c goes asymptotically to one and most of the traffic is served by the same first stage queue. Consequently, for a fixed α, $S_1(t)$ increases with N. Equivalently, $S_h(t)$ decreases.

In the limit, we can approximate the model by just two constant service time queues as shown in Fig. 11(b), where no buffer is needed in the head node. ■

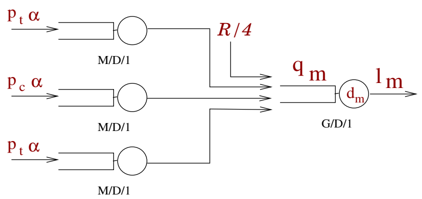
Lemma 2: Row-first routing is also optimal among the shortest path space invariant routing algorithms for finite buffer networks.

Proof: First, to minimize overflow losses, it is necessary to minimize λ_{q_m}. From Theorems 1 and 2, row-first routing generates the minimum λ_{q_m} for both torus and square lattices. Moreover, we have seen in Theorem 5 that packets are stored in the buffer mainly due to turning traffic (p_t) and by reducing this traffic, the number of packets in q_m is also reduced. Note that for any source-destination pair, row-first routes packets turning the minimum number of times. That is, among all the routing algorithms that generates the minimum λ_{q_m}, row-first generates the minimum turning traffic (p_t). Consequently, row-first generates the minimum number of packets in q_m. ■

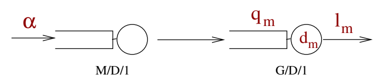
We can simplify our model even further while still keeping the important properties that determine the queue size distribution. Since p_t is $\mathcal{O}(1/N)$, we can reduce the model for large networks by assuming that the number of packets turning at d_m is negligible. That is, the packets arrive at q_m only from the neighbor located in the same row or column as l_m. Similarly, the exogenous input traffic generated at d_m, goes asymptotically to zero ($\mathcal{O}(1/N)$) compared with the incoming traffic α and can also be neglected.

Consequently, we approximate the queue network by a two-queue model where q_m is a deterministic service time queue that receives traffic from another deterministic service time queue with the same service time and average input traffic equal to α (Fig. 11(b)). It follows that the number of packets in q_m is (at most) one with probability α and zero with probability $1-\alpha$.

Finally, the distribution $P_m(k)$ on the total queue size k at d_m, is given by the addition of four

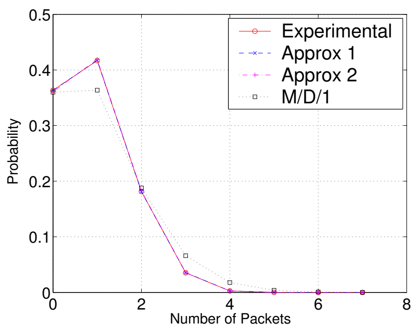


(a)

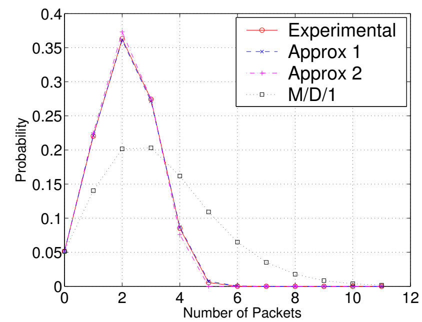


(b)

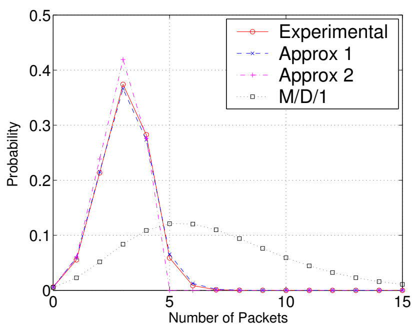
Fig. 11. Approximation models. (a) Two-stage model used to analyze the distribution on the size of q_m. (b) Two-queue model: reduced two-stage model without crossing packets and without considering input traffic from d_m.



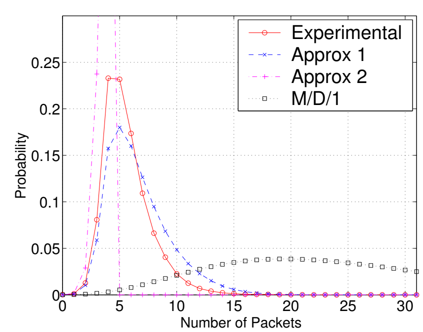
(a) α = 0.225



(b) α = 0.525



(c) α = 0.725



(d) α = 0.925

Fig. 12. Distribution on the queue size at d_m for different values of α in a 121×121 square lattice network with full-duplex links.

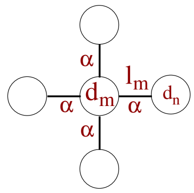
independent and identically distributed queues associate to the four outgoing links from d_m:

$$P_m(k) = \begin{cases} \binom{4}{k}(1-\alpha)^{(4-k)}\alpha^k, & \text{for } 0 \le k \le 4, \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

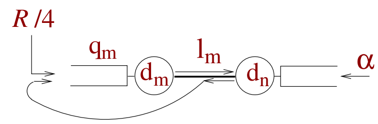
Fig. 12 shows the distribution on the size of q_m for different values of α in a 121×121 square lattice network. This figure compares the different distributions obtained by simulation, the two-stage model (Fig. 11(a)), the two-queue model (Fig. 11(b)) and the usual M/D/1 approximation.

For the M/D/1 approximation, we simply apply Jackson's Theorem and consider that each queue in the network is M/D/1 and independent of other queues [18]. Therefore, we approximate q_m by a M/D/1 queue with a Poisson arrival with rate α.

Both the two-stage model and two-queue model closely approximate the experimental distribution for



(a)



(b)

Fig. 13. Half-duplex approximation model. (a) Network with half-duplex links. (b) Approximated model for the queue associated to l_m.

low to moderate rates. Experimentally, we have found that a good approximation is obtained for $\alpha < 0.8$. Beyond this traffic intensity, some of the assumptions we make are not totally valid and the approximation quality degrades. We neglected the traffic leaving the network at each node to apply Theorem 4. Since the average leaving traffic per node is equal to $\mathcal{R}$, this approximation is less accurate as we increase the rate per node.

Furthermore, both models become closer to the experimental distribution as the size of the network N increases. Note that our two approximation models become asymptotically exact.

Fig. 16(a) shows the distribution on the queue size at d_m for a constant relative capacity $\alpha = 0.75$ and different network sizes N. This plot confirms that, as N increases, the packet distribution converges to the two tandem queues model (Fig. 11(b)). Consequently, as stated in Theorem 5, the probability of having more than four packets in d_m (one for each output link l_m) goes to zero as we increase the network size.

2) Half-Duplex communication channels: For half-duplex channels, we cannot apply the same techniques as in the full-duplex case since the arrival and service times in d_m are no longer independent. If d_m receives k packets from its neighbors, not only its queue is increased by k packets, but also it can transmit, at most, $4 - k$ packets using the remaining links.

We assume as before that d_m is composed of four independent and identically distributed queues associated to the four output links and we analyze distribution on the size of one of these queues q_m associated to the output link l_m.

To capture the dependence between arrivals and departures, we propose the following approximation model. Every time d_m wants to send a packet through l_m, it has to compete for l_m with one of its neighbors, d_n (Fig. 13(a)). If d_m takes l_m first, it can transmit a packet and the size of q_m is reduced by one. However, if d_n takes the link first and sends a packet, not only d_m is unable to transmit, but also the size of q_m is increased by one if the final destination of the packet is not d_m.

Note that, in practice, packets sent by d_n never go through l_m (packets do not go backwards) although they stay in d_m. However, by putting these packets into l_m we simulate packets arriving from the other neighbors of d_m and prevent packet transmissions. This approximation is represented in Fig. 13.

We denote by ρ_m the utilization factor of q_m. That is,

$$\rho_m = \frac{\lambda_{q_m}}{\mu_{q_m}} = \frac{\alpha}{\mu_{q_m}},$$

where λ_{q_m} is the arrival rate to q_m, and μ_{q_m} is the service rate. Note that λ_{q_m} is identical in both half-duplex and full-duplex models.

Similarly, we denote by ρ_n the utilization factor of the queue q_n in d_n associated to l_m. We assume that the probability that d_m captures the link before d_n is equal to $1/2$. Therefore, if q_m has a packet waiting to be transmitted, the probability p_s of sending it this time slot is simply equal to the probability of d_m being the first to capture the link plus the probability of d_n having nothing to transmit through l_m:

$$p_s = \frac{1}{2} + \frac{1}{2}(1 - \rho_n) = 1 - \rho_n/2; \tag{24}$$

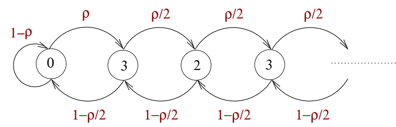
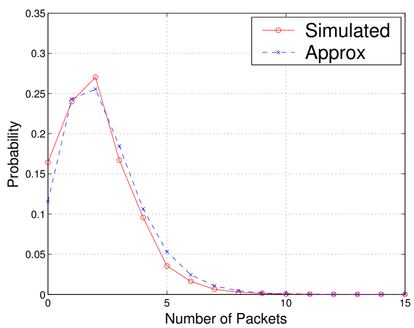
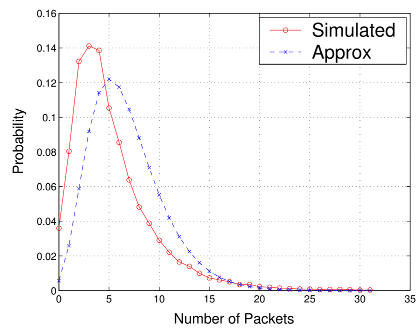


Fig. 14. Markov chain model for the half-duplex links model



(a) α = 0.725



(b) α = 0.925

Fig. 15. Distribution on the queue size at d_m for different values of α in a 121×121 square lattice network with half-duplex links.

We model q_m service time as a geometric distribution with parameter p_s. That is, if d_m does not capture l_m in this time slot, we assume that it tries to capture it in the next time slot with the same probability.

As in the full-duplex case, we approximate arrivals to d_n as a Poisson distribution with parameter α. Accordingly, interarrival times are independent and exponentially distributed with the same parameter.

In addition to arrivals from d_n, new packets are also produced at d_m following a Bernoulli distribution with rate $\mathcal{R}$. Considering again the symmetry of d_m, the fraction of this traffic that goes through l_m is $\mathcal{R}/4$.

Note that both distributions, arrivals and service time, are memoryless. This memoryless condition allows to use a Markov chain analysis, that is, if we denote by $X_m(t)$ the number of packets in the queue q_m at time t, $\{X_m(t) \mid t > 0\}$ can be approximated using a Markov chain.

As the network size increases, the difference between both utilization factors ρ_m and ρ_n becomes negligible, and we can assume that $\rho_m = \rho_n = \rho$. Moreover, the new traffic generated at d_m becomes negligible ($\mathcal{O}(1/N)$) compared to the traffic that arrives from d_n.

Applying these simplifications, the transition probability matrix $P_m(j,k)$ associated to $\{X_m(t) \mid t > 0\}$ can be approximated by:

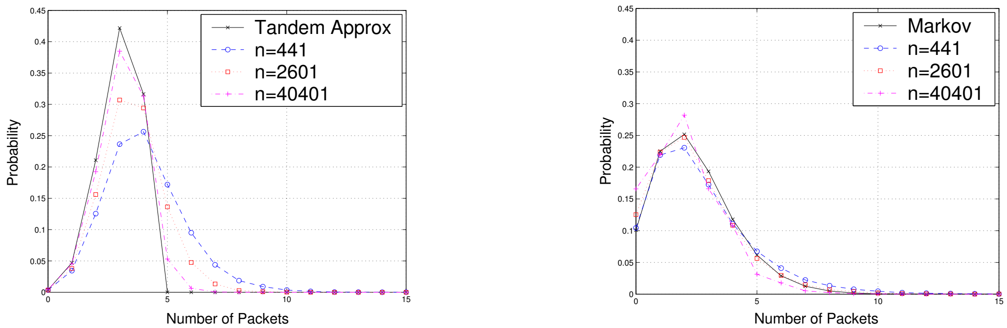
$$P_m(0,k) = \begin{cases} 1-\rho, & k=0, \\ \rho, & k=1, \end{cases} \tag{25}$$

$$P_m(j,k) = \begin{cases} 1-\rho/2, & k=j-1, \\ \rho/2, & k=j+1, \end{cases} \tag{26}$$

whose transition graph is shown in Fig. 14.

Fig. 15 shows the distribution on the queue size at d_m for different values of α in a 121×121 square lattice network with half-duplex links. This figure compares the packet distribution obtained by simulation with the Markov chain approximation.

This model closely approximates the experimental distribution for low to moderate rate per node ($\alpha < 0.8$). As in the case of full-duplex links, beyond this traffic intensity, some assumptions are no more valid and the approximation quality degrades.



(a) full-duplex (b) half-duplex

Fig. 16. Distribution on the queue size at d_m for a constant relative capacity $\alpha = 0.75$ and different network sizes with (a) full-duplex and (b) half-duplex links.

Fig. 16(b) shows the distribution on the queue size at d_m for a constant relative capacity $\alpha = 0.75$ and different network sizes in the case of half-duplex links.

A key difference with the case of full-duplex links is that, as the network size increases, the buffer requirements do not go asymptotically to zero. The intuitive reason is that, in the case of half-duplex links, l_m is shared between d_m and d_n and, even if the input rate λ_{l_m} is less than the link capacity, there is a non-zero probability that d_m compete for the link with d_n, in which case one of them has to store the packet for a further transmission. In other words, the stationary probability distribution $\nu_m(k)$ of the Markov chain that approximates the number of packets k in q_m, has positive values for $k > 1$ for any value of N.

V. Data Gathering Model

In this section, using a similar methodology as before, we analyze the data gathering model. In data gathering, every node transmits information to a particular and previously designated node d_{BS} denoted *base station* that can be located anywhere in the network. The communication matrix T is given by:

$$T(d_m, d_j) = \begin{cases} 1, & \text{if } d_j = d_{BS}, \\ 0, & \text{otherwise.} \end{cases}$$

In data gathering, routing in a torus is a particular case of routing in a square lattice. The reason is that for any base station location in a torus, the shortest path graph consists of a square lattice with the base station located in the center node. Therefore, in this section we only consider the square lattice network.

We apply the same type of analysis as in the uniform communication model. First, we bound the network capacity $C_s^{dg}(N)$ and analyze optimal routing algorithms that achieve this capacity under the infinite buffer assumption. Second, we analyze the effect of finite buffers.

A. Network Capacity

The following Lemma establishes an upper bound for the network capacity under the data gathering model:

Lemma 3: The network capacity $C_s^{dg}(N)$ for the data-gathering communication model in a square lattice is upper bounded as follows:

$$C_s^{dg}(N) \leq \frac{|\varphi(d_{BS})|}{N^2 - 1}. \tag{27}$$

Proof: Since all the traffic from the network must reach d_{BS} using one of the links in the set $\varphi(d_{BS})$, the bottleneck of the network is clearly located in these links. Applying a bisection argument [19] to these links yields the result. ∎

Note that, in the links that limit the capacity, the information flows only in one direction: from the inner nodes to d_{BS}. Therefore, the network capacity is equivalent for half-duplex and full-duplex links.

B. Optimal Routing Algorithms for Infinite Buffers

We define λ_l^{Π} as the average arrival rate to link l according to a routing algorithm Π. Capacity achieving routing algorithms, under the infinite buffer hypothesis, are characterized by the following Lemma:

Lemma 4: A shortest path routing algorithm Π achieves capacity for a location of the base station d_{BS} if and only if the total arrival traffic to d_{BS} is uniformly distributed among the links in $\varphi(d_{BS})$. That is:

$$\lambda_l^{\Pi} = \frac{\mathcal{R}(N^2-1)}{|\varphi(d_{BS})|}, \quad \text{for all } l \in \varphi(d_{BS}). \tag{28}$$

Proof: Since all the arriving traffic to d_{BS} has to use one of the links in the set $\varphi(d_{BS})$,

$$\sum_{l \in \varphi(d_{BS})} \lambda_l^{\Pi} = \mathcal{R}(N^2-1). \tag{29}$$

For data gathering model, the most loaded link in the network obviously belongs to $\varphi(d_{BS})$. Therefore:

$$\max_{l \in \varphi(d_{BS})} \lambda_l^{\Pi} > \lambda_{l_x}^{\Pi} \quad \text{for all } l_x \in E_s \setminus \varphi(d_{BS}).$$

The stability condition in the links is given by $\rho = \frac{\lambda_l}{\mu_l} < 1$. As we consider unitary capacity links,

$$\max_{l \in \varphi(d_{BS})} \lambda^{\Pi}(l) < 1 \quad \text{for all } l \in \varphi(d_{BS}). \tag{30}$$

Combining (29) and (30), the result follows. ∎

As a consequence of Lemma 4, we have the following achievability result:

Corollary 5: The network capacity $C_s^{dg}(N)$ of a square lattice network for data gathering model is equal to the upper bound given by (27).

C. Routing with Finite Queues

Lemma 4 establishes that the only necessary and sufficient condition for a routing algorithm to achieve capacity under the infinite buffer assumption is to uniformly distribute the traffic among the four links of d_{BS}. Although there is a wide class of routing algorithms that satisfy this condition, we show in this section that their performance is quite different when the buffers are constrained to be finite.

For the sake of simplicity, we restrict our analysis to a particular location of d_{BS}: the square lattice center. Note that the analysis of this location is equivalent to solving the problem for any location in the torus network. Nevertheless, a similar analysis can be carried out for any location.

Note that, for the central node, row-first routing does not satisfy the optimality condition: by always forwarding packets along the same row until they reach the column of d_{BS}, most of the traffic reach d_{BS} through the upper and lower links while the rest of the links are underused. However, there are many routing algorithms that achieve capacity under the infinite buffer assumption. For instance, a simple routing algorithm that satisfies the capacity condition is the *random greedy algorithm* [19]. In this case, nodes use row-first or column-first as routing algorithm with equal probability.

To analyze the network capacity for a given routing algorithm under finite buffers, we proceed as in the uniform communication model. First, we identify the most loaded node d_m and associate the network to a tree. Then, we reduce it to its two-stage model and obtain the packet distribution in d_m by analyzing

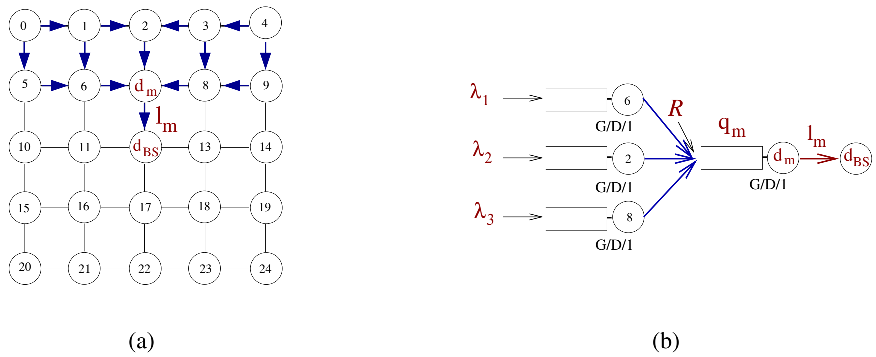


Fig. 17. Two-stage equivalent model. (a) Tree network where l_m is the head node and (b) its two-stage model.

the packet distribution in the head node of the two-stage model. We perform this analysis for any shortest path routing algorithm Π that achieves capacity under the infinite buffer assumption.

The bottleneck of the network is clearly located in the four neighbors of d_{BS}. Moreover, according to Lemma 4, the total arrival traffic to d_{BS} is uniformly distributed among the links in $\varphi(d_{BS})$. Due to the independence of packet generation, the distributions on the queue size in these four nodes are independent and identically distributed. Consequently, we reduce the problem to computing the queue distribution for one of these neighbors d_m. We denote by l_m the link between d_m and d_{BS} and by q_m the queue in d_m associated to l_m (Fig. 17(a)).

We consider now only those nodes that generate traffic through l_m. These nodes form a tree with q_m as head, with exogenous inputs at each node, and with no traffic leaving the network. Applying Theorem 4, the packet distribution in q_m is the same as in its two-stage model (Fig. 17(b)). Note that in this case the two-stage model is not an approximation (as in the uniform communication model) but an exact model for any rate.

The arrivals to the three nodes of the first stage are the sum of all the traffic generated by the network that goes through l_m. If Π achieves capacity for infinite buffers, by Lemma 4, the total average traffic that flows through l_m is equal to:

$$\lambda_{l_m} = \frac{\mathcal{R}(N^2-1)}{4}. \tag{31}$$

We denote by λ_1, λ_2 and λ_3 the average arrival rates to the three first stage nodes of the two-stage model (Fig. 17(b)). These three nodes have to route all the traffic that goes through l_m except the traffic generated by d_m itself. That is,

$$\lambda_1 + \lambda_2 + \lambda_3 = \frac{\mathcal{R}(N^2-1)}{4} - \mathcal{R}. \tag{32}$$

We obtain the distribution on the size of q_m by analyzing the distribution at the head node of the two-stage model. This distribution obviously depends on the values of λ_1, λ_2 and λ_3.

We are interested in finding the routing algorithm Π that achieves the maximum rate per node $\mathcal{R}^{\Pi}_{\text{max}}(N,Q)$. Maximizing $\mathcal{R}^{\Pi}_{\text{max}}(N,Q)$ is equivalent to minimizing the number of packets in d_m, that is, in q_m.

Different routing algorithms generate different values for λ_1, λ_2 and λ_3, and consequently, different distributions on the size of q_m. First, we analyze the values of λ_1, λ_2, and λ_3 that generate the minimum number of packets in q_m and then, we analyze the routing algorithm that induces such values.

Lemma 5: In a two-stage network where the total average arrival rate is fixed, i.e., $\lambda_1 + \lambda_2 + \lambda_3 = \lambda_t$, the values of λ_i that minimize the number of packets in the head node for any arrival distribution are such that all traffic arrives only through one node of the first stage. That is:

$$\lambda_i = \begin{cases} \lambda_t, & \text{for } i\text{=1,2 or 3,} \\ 0, & \text{otherwise.} \end{cases}$$

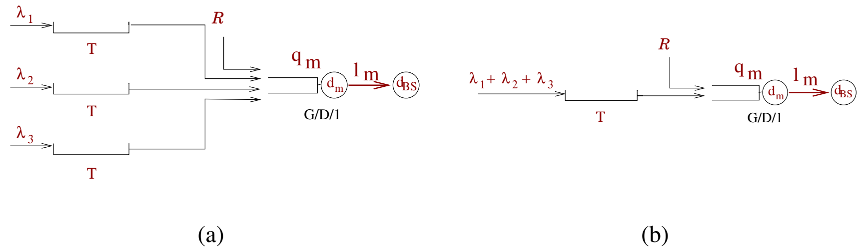


Fig. 18. Two-stage equivalent model. (a) Two-stage model where the first stage queues has been replaced by pure delays of T time slots, (b) equivalent to injecting all the arrivals to a single pure delay.

Proof: By Theorem 3, the total number of packets in the two-stage model (Fig. 17(b)) is the same as a system where the first stage queues has been replaced by pure delays of T time slots (Fig. 18(a)). In terms of number of packets in the system, this is equivalent to injecting all the arrivals to a single pure delay (Fig. 18(b)). Consequently, the total number of packets in the system is equivalent for any combination of λ_i values.

We can decompose the number of packets in the two-stage model as the packets in the first stage plus packets in the head node. Minimizing the number of packets in the head node is therefore equivalent to maximizing the packets in the first stage.

Since the first stage is composed of three $G/D/1$ queues with equal service time, the number of packets in the first stage is maximized when all the traffic goes through only one queue. Equivalently, the number of packets in the head node is minimized. ■

As a consequence of Lemma 5, the routing algorithm that achieves the maximum $\mathcal{R}^{\Pi}_{\max}(N,Q)$ is such that the input traffic to d_m arrives only from one of its neighbors.

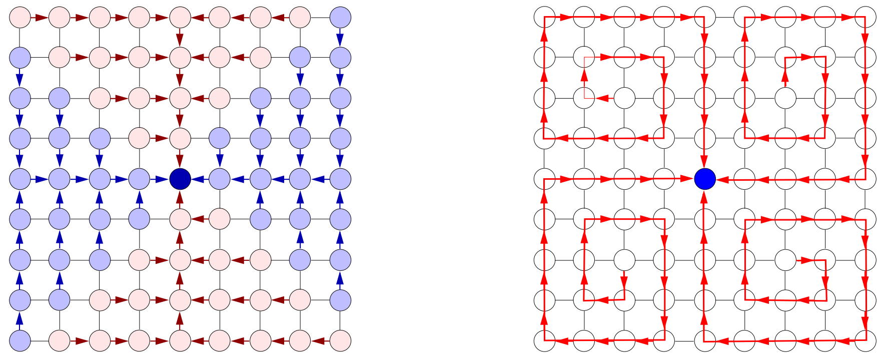
However, the congestion problem is now translated to this neighbor of d_m. Furthermore, as the network size increases, the difference between the traffic that flows through d_m and its neighbor goes asymptotically to zero. Therefore, we have to apply Lemma 5 recursively. That is, the optimal routing algorithm is such that nodes receive most of their traffic from only one neighbor.

The shortest path routing algorithm that implements this principle is shown in Fig. 19(a) and it is as follows. In the $N \times N$ square lattice, there are $2(N-1)$ nodes that only have one possible shortest path toward d_{BS}. We denote this set of nodes by $SD(d_{BS})$. For any other node, the optimal routing algorithm consists of forwarding packets to the closest node in $SD(d_{BS})$. Note that there is only one closest node in $SD(d_{BS})$ for all the nodes except for those nodes located in the two diagonals of the square lattice. Diagonal nodes forward packets only towards one of the two closest nodes in $SD(d_{BS})$ in such a way that each of the four diagonal nodes at the same distance from d_{BS} chooses a different node. We denote this routing algorithm as *cross routing*.

Among all shortest path routing algorithms, cross routing generates the optimal node arrival distribution according to Lemma 5. Although the nodes that support more traffic receive packets from more than one neighbor, most of the traffic arrives mainly from one. The average arrival rates generated by cross routing in d_m are $\lambda_1 = \mathcal{R}(N^2-9)/4$, $\lambda_2 = \mathcal{R}$, and $\lambda_3 = 0$. It follows that cross routing is asymptotically optimal.

According to Lemma 5, the optimal routing consists on making nodes receive all traffic exclusively from one neighbor. This condition can only be fully satisfied by a *non-shortest path* routing. Applying again this condition recursively, the optimal routing algorithm consists on making traffic flow toward d_{BS} following a spiral as shown in Fig. 19(b). We denote this routing algorithm as *snail routing*. Snail routing generates the optimal arrival distribution in all nodes.

Although the snail routing achieves the maximum $\mathcal{R}^{\Pi}_{\max}(N,Q)$, the delay incurred by the packets may be unacceptable. Notice that a packet generated by the furthest node must travel across $(N^2-1)/4$ nodes before reaching d_{BS}, while for a shortest path routing, the furthest node is $N-1$ hops away. Moreover, the average path length $\overline{L}_{snail}$ for snail routing is $\mathcal{O}(N^2)$ while for any shortest path routing, $\overline{L}_{s-p}$ is



(a) Cross routing (b) Snail routing

Fig. 19. Routing algorithms for finite buffers. (a) Cross routing, the optimal shortest path routing and (b) Snail routing, the optimal non-shortest path data gathering routing.

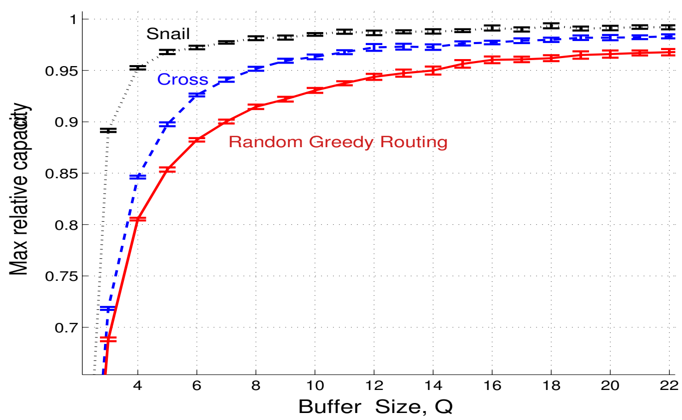


Fig. 20. Routing for data gathering: maximum rate $\mathcal{R}^{\Pi}_{\max}(N,Q)$ achieved by different routing algorithms in a 21×21 square lattice for different buffer sizes Q with the 95% confidence intervals.

$\mathcal{O}(N)$. Snail routing represents an extreme case of the existing trade-off between $\mathcal{R}^{\Pi}_{\max}(N,Q)$ and delay: achieving the optimal rate per node drastically increases the delay.

Fig. 20 shows the $\mathcal{R}^{\Pi}_{\max}(N,Q)$ achieved by different routing algorithms in a 21×21 square lattice network as a function of the buffer size Q with the 95% confidence intervals. We compare experimentally the performance of random greedy routing, cross routing and snail routing.

Notice that although all routing algorithms asymptotically achieve capacity as the buffer size increases, the maximum achievable rate per node $\mathcal{R}^{\Pi}_{\max}(N,Q)$ under small buffers differs strongly among different routing algorithms. As expected, the maximum $\mathcal{R}^{\Pi}_{\max}(N,Q)$ corresponds to snail routing, while cross routing performs best among shortest path routing algorithms.

Fig. 21 shows the maximum rate achieved by different routing algorithms relative to the maximum rate achieved by the snail routing for a fixed buffer size $Q = 5$ as a function of the network size N with the 95% confidence intervals. Since all routing algorithms analyzed are asymptotically optimal with the network size, the performance gap between snail routing and these algorithms decreases as the network size increases for a fixed value of Q. The reason is that, as the network size increases, the most loaded nodes receive most of the traffic mainly from only one neighbor.

D. The broadcast model

We can easily apply the analysis shown in this section to the analogous broadcast communication model. Suppose now that the central node is the only source in the network and generates different information

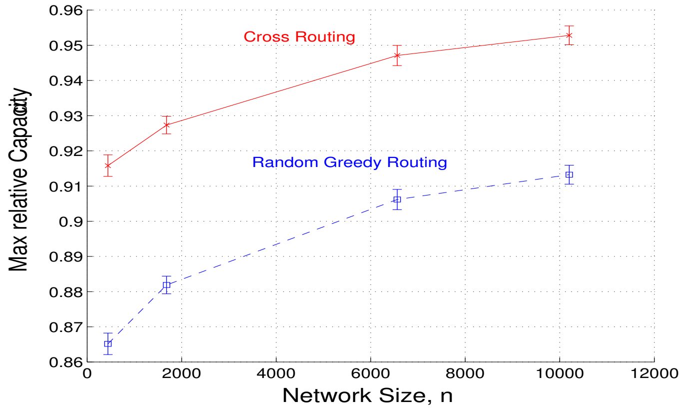


Fig. 21. Routing for data gathering: performance of cross routing and greedy routing relative to snail routing for a fixed buffer size $Q = 5$ for different network sizes n with the 95% confidence intervals.

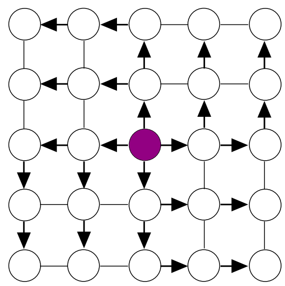


Fig. 22. Routing algorithm for broadcast communication model.

packets for each node of the network. Using the same reasoning as before, the optimal routing algorithm is such that the output traffic from the central node is uniformly distributed among its four output links, and all the nodes receive traffic from only one of its neighbors, which is optimal in the sense of finite queues. It is important to note that in this case, since the most loaded nodes (nodes closer to the central node) always receive traffic from only one of its neighbors (the central node), the differences in performance between routing algorithms is not as significant as in the data gathering model. Many different routing algorithms can be provided with similar performance, for instance, the routing algorithms illustrated in Fig 22.

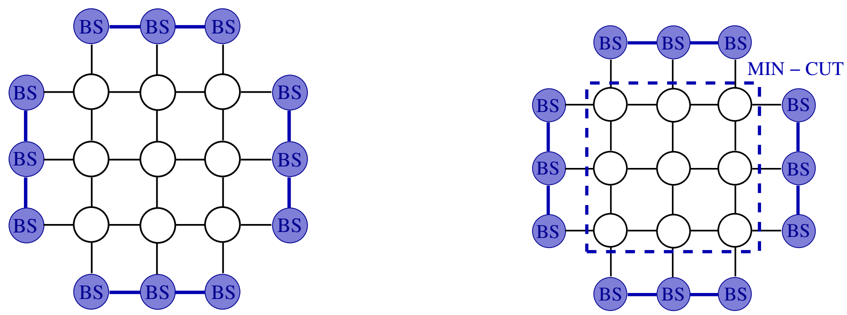
VI. Border Data Gathering Model

In this section we apply similar tools to analyze routing in a substantially different communication model, namely, border data gathering. In border data gathering, all nodes located in the four edges of the square lattice act as base stations and inner nodes act as sources generating information that needs to be transmitted to any of these base stations without any specific mapping between source nodes and base stations (Fig. 23(a)). Therefore, several communication matrices are allowed. Obviously, this model can be considered only for the square lattice.

We proceed as in previous sections: first, we compute the network capacity with infinite buffer queues based only on flow arguments. Then, we present the set of routing algorithms that achieve capacity under the infinite buffer assumption. Finally, we consider finite buffers and analyze the routing policies that maximize the achievable rate per node for a given buffer size Q.

A. Network Capacity

The following Lemma establishes an upper bound for the network capacity under the border data gathering communication model:



(a) (b)

Fig. 23. Border data gathering model. (a) Inner nodes generate packets to be transmitted to any of the base stations located in the square lattice edges. (b) Bisection that determines the network capacity.

Lemma 6: The network capacity $C_s^{bg}(N)$ for the border data gathering communication model in a square lattice is upper bounded as follows:

$$C_s^{bg}(N) \leq \frac{4}{N-2}. \tag{33}$$

Proof: To determine the network capacity $C_s^{bg}(N)$, we apply again the bisection argument. The bottleneck of the network is clearly determined by the links connecting inner nodes to base stations. Therefore, the bisection that determines the network capacity is the one that separates the edge nodes from inner nodes (Figure 23(b)). Noticing that there are $(N-2)^2$ nodes and $4(N-2)$ links through the cut, the result follows. ■

Note also that, in the links that limit the capacity, the information flows only in one direction: from the inner nodes to the edge nodes. Therefore, the network capacity is equivalent for half-duplex and full-duplex links.

B. Optimal Routing Algorithms for Infinite Buffers

Let S_{BS} denote the set of base stations in the network and $\varphi(S_{BS})$ the set of links that connect any base station with an inner node. Lemma 6 establishes that the maximum rate that nodes can transmit to S_{BS} is determined by the number of links in $\varphi(S_{BS})$. As a consequence, capacity achieving routing algorithms under the infinite buffer hypothesis, are characterized by the following Lemma:

Lemma 7: A routing algorithm Π achieves capacity only if the total arrival traffic to S_{BS} is uniformly distributed among the links in $\varphi(S_{BS})$. That is:

$$\lambda_l^{\Pi} = \frac{\mathcal{R}(N-2)}{4}, \qquad \text{for all } l \in \varphi(S_{BS}). \tag{34}$$

Since the proof of this lemma is similar to the proof of Lemma 4, we omit the proof here.

First of all, notice that no shortest path routing algorithms satisfies exactly this optimality condition. The best shortest path routing algorithm consists on distributing the traffic as uniformly as possible among the links in $\varphi(S_{BS})$. Consequently, when a node has more than one possible shortest path toward a base station, it distributes the load uniformly among these paths. This optimal shortest path routing is shown in Fig. 24(a). The maximum rate $\mathcal{R}_{\text{max}}^{\text{s-p}}(N,\infty)$ achieved by the optimal shortest path routing is limited by the most loaded links, that is, the links located in the middle of the four edges (Fig. 24(a)):

$$\mathcal{R}_{\text{max}}^{\text{s-p}}(N,\infty) = \frac{4}{2N-5} < C_s^{bg}(N),$$

which is roughly one half of the network capacity $C_s^{bg}(N)$.

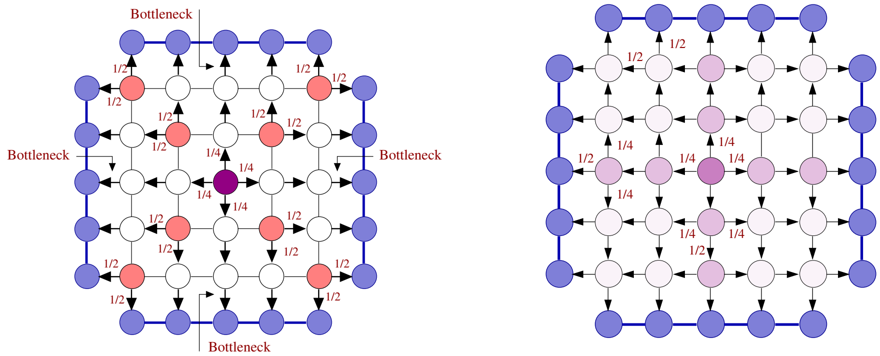


Fig. 24. Routing algorithms for border data gathering. (a) Optimal shortest path routing. (b) Uniform data gathering.

On the other hand, it can be shown that there exist multiple non-shortest path routing algorithms that achieve capacity. For instance, a simple strategy that achieves capacity consists on the following: packets are always routed along the same row or column until they reach a base station, that is, packets do not turn. New packets generated in a node are routed with equal probability to the two closest base stations located in the same row or column as the node. If there are more than one base stations at the same distance, it chooses any of them with equal probability. We denote this routing algorithm as *uniform data gathering* and it is depicted in Fig. 24(b). Uniform data gathering satisfies the capacity condition (34) and therefore,

$$\mathcal{R}^{\text{uniform}}_{\text{max}}(N,\infty) = C_s^{bg}(N).$$

Note that since in both routing algorithms, shortest path and uniform data gathering, packets flow only in one direction, the routing algorithms are equivalent for half-duplex and full-duplex links.

C. Routing with Finite Buffers

To derive the routing algorithm that achieves the maximum $\mathcal{R}^{\Pi}_{\text{max}}(N,Q)$ for finite queues, we apply the condition we derived in section V-B. That is, the optimal routing policy is such that nodes receive most of their traffic from only one of its neighbors.

Note that the optimal shortest path routing algorithm for infinite buffers is also optimal for finite buffers since all nodes receive traffic from only one neighbor. However, this routing policy does not achieve capacity with infinite buffers. On the other hand, although uniform data gathering achieves capacity for infinite buffers, most nodes receive traffic from many of its neighbors, and therefore, it does not satisfy the optimality condition for finite buffers.

Actually, as we shown in the next Lemma, in border data gathering no routing algorithm reaches capacity and behaves optimally under finite buffers.

Lemma 8: In border data gathering, the optimal queue condition for finite buffers that minimizes the number of packets in the queues, and the capacity condition (34) cannot be both satisfied.

Proof: Let Π be a routing algorithm that achieves capacity under the infinite buffer assumption. Consider a node d_a located in the diagonal close to the edges, as illustrated in Fig. 25(a). Notice that it is enough to focus ourselves on the traffic that goes through a certain node. By Lemma 7, d_a has to carry the traffic of at least two links in $\varphi(S_{BS})$, that is, $\lambda_{d_a} \geq 2\frac{\mathcal{R}(N-2)}{4}$. Consequently, d_a receives traffic from

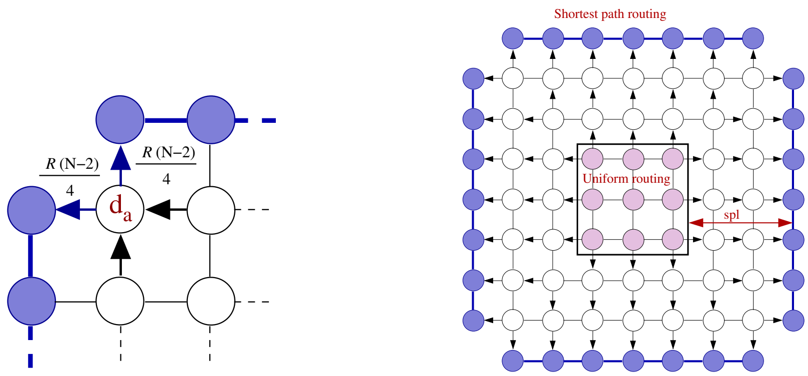


Fig. 25. Maximum rate trade-off in border data gathering: (a) the nodes located in the diagonal close to the edges carry the traffic of at least two bottleneck links. (b) Adaptive routing.

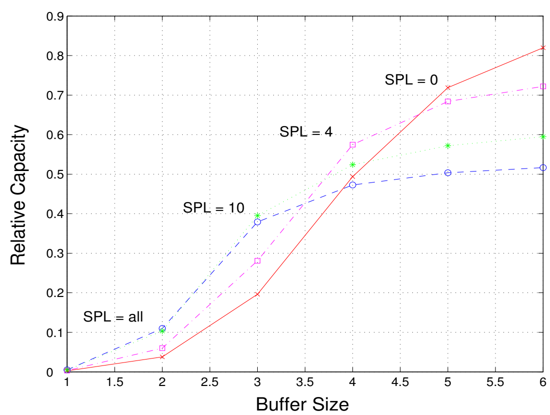


Fig. 26. Adaptive routing: maximum relative capacity achieved by adaptive routing with different shortest-path limit values as a function of the buffer size in a 41×41 square lattice.

more than one neighbor. Otherwise, there would be a link l such that $\lambda_l > \frac{\mathcal{R}(N-2)}{4}$, and Π would not achieve capacity. On the other hand, if d_a receives traffic from only one neighbor, λ_{d_a} is upper bounded as $\lambda_{d_a} \leq \frac{\mathcal{R}(N-2)}{4}$, and (34) is not satisfied. ■

Lemma 8 implies that the design of the optimal routing algorithm for a given queue size Q, should trade-off both the conditions related to approaching capacity and the conditions related to operate optimally in the case of finite queues. Therefore, we propose an adaptive routing algorithm that depends on the buffer size Q. As in previous sections, the most critical nodes are those located close to the base stations, that is, the most loaded nodes. Therefore, it is in those nodes where it is more important to apply the optimal queue condition.

Consequently, we define the following routing algorithm: nodes located at a distance less than a fixed value *spl* (shortest path limit) from any base station, route packets according to shortest path routing. Nodes further than spl route packets according to the uniform data gathering. We denote this routing algorithms as *adaptive routing* and it is depicted in Figure 25(b). Note that when spl is equal to zero, adaptive routing is equivalent to uniform data gathering, that is, the more loaded node (nodes close to the border) receives traffic from more than one neighbor. As we increase the value of spl, these nodes start receiving packets from only one neighbor (shortest path routing). Finally, when spl is equal to $(N-1)/2$, all nodes receive packets from only one neighbor and adaptive routing is equivalent to shortest path routing.

Fig. VI-C shows the values of $\mathcal{R}^{\Pi}_{\max}(N,Q)$ achieved by adaptive routing algorithm with different values of spl in a 41×41 square lattice network, as a function of the buffer size Q. First, note the trade-off between the rate $\mathcal{R}^{\Pi}_{\max}(N, Q)$ achieved for big and small buffer sizes: no routing strategy can achieve high rates for both extremes. For high buffer values, the optimal routing strategy consists on choosing spl=0, which results in uniform routing. As the buffer size decreases, the optimal values for spl decrease and, when the buffer goes to zero, the optimal value for spl is the maximum, which results in shortest path routing.

VII. Conclusions

In this paper, we studied the problem of routing in lattice networks with infinite and finite buffers under three different communication models, the uniform model, the data gathering model, and the border data gathering model. We presented alternative approximation models to the usual Jackson's Theorem that allowed us to obtain a more accurate distribution on the queue size at the most loaded node, that is, the rate-limiting node.

Using these approximation models, we have proposed a simple rule to design routing algorithms that achieve the highest maximum rate per node in the case where nodes have a finite buffer. This rule consist on making nodes receive most of their traffic from only one of its neighbors. We have applied this rule to different communication models, namely data gathering and border data gathering, to design the optimal shortest path and non-shortest path routing algorithms for nodes with a finite buffer.

References

[1] G. Barrenetxea, B. Beferull-Lozano, and M. Vetterli, "Lattice sensor networks: Capacity limits, optimal routing and robustness to failures," in *International Workshop on Information Processing in Sensor Networks (IPSN)*, April 2004.

[2] J. H. Upadhyay, V. Varavithya, and P. Mohapatra, "Efficient and balanced adaptive routing in two-dimensional meshes," in *HiPC96, Proceeding of High Performance Computing*, 1996, pp. 112–121.

[3] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 9, pp. 1478–1495, Sep. 2003.

[4] A. El Gamal, "Trends in cmos image sensor technology and design," in *IEEE International Electron Devices Meeting*, Dec. 2002.

[5] D. Bhatia, T. Leighton, F. Makedon, and C. H. Norton, "Improved algorithms for routing on two-dimensional grids," *Lecture Notes in Computer Science*, vol. 657, pp. 114–??, 1993.

[6] J. F. Sibeyn, "Overview of mesh results," Technical Report MPI-I-95-1-018, Max-Planck-Institut für Informatik, Saarbrücken, 1995.

[7] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: Mobile networking for "smart dust"," in *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom-99)*, Aug. 1999, pp. 271–278.

[8] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, May 2000.

[9] Crossbow Products: Wireless sensor networks, "http://www.xbow.com/products/new_product_overview.htm," .

[10] Y. Liu, M. T. Hill, H. de Waardt, G. D. Khoe, and H. J. S. Dorren, "All-optical buffering using laser neural networks," *IEEE Photonics Technology Letters*, vol. 15, no. 4, pp. 596–598, April 2003.

[11] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall International Editions, 1992.

[12] M. D. Grammatikakis, D. F. Hsu, M. Kraetzl, and J. F. Sibeyn, "Packet routing in fixed-connection networks: A survey," *Journal of Parallel and Distributed Computing*, vol. 54, no. 2, pp. 77–132, 1 Nov. 1998.

[13] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[14] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Info. Theory*, vol. 46(2), pp. 388–404, March 2000.

[15] R. Duncan, "A survey of parallel computer architectures," *IEEE Computer*, vol. 23, no. 2, 1990.

[16] N. F. Maxemchuk, "Comparison of deflection and store-and-forward techniques in the Manhattan street and shuffle-exchange networks," in *INFOCOM, Eighth Annual Joint Conference of the IEEE Computer and Communications Societies*, 1989, vol. 3.

[17] M. Harchol-Balter and P. Black, "Queueing analysis of oblivious packet-routing networks," in *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, Daniel D. Sleator, Ed., Arlington, VA, Jan. 1994, pp. 583–592, ACM Press.

[18] M. Mitzenmacher, "Bounds on the greedy routing algorithm for array networks," in *Proceedings of the 6th Annual Symposium on Parallel Algorithms and Architectures*, New York, NY, USA, June 1994, pp. 346–353, ACM Press.

[19] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures*, Morgan-Kaufman, 1991.

[20] M. J. Neely and C. E. Rohrs, "Equivalent models and analysis for multi-stage tree networks of deterministic service time queues," in *38th Annual Allerton Conference on Communication, Control and Computing*, Oct. 2000.

[21] M. J. Neely, C. E. Rohrs, and E. Modiano, "Equivalent models for analysis of deterministic service time tree networks," Tech. Rep. P-2540, MIT - LIDS, March 2002.

[22] G. Barrenetxea, B. Beferull-Lozano, A. Verma, P.L. Dragotti, and M. Vetterli, "Multiple description source coding and diversity routing: a joint source channel coding approach to real-time services over dense networks," in *Proc 13th Int. Packet Video Workshop*, April 2003.