

Lattice Networks: Capacity Limits, Optimal Routing, and Queueing Behavior

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

Abstract—Lattice networks are widely used in regular settings like grid computing, distributed control, satellite constellations, and sensor networks. Thus, limits on capacity, optimal routing policies, and performance with finite buffers 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 on the queue distribution and derive the algorithms that achieve the maximum rate.

Index Terms—Border data gathering, data gathering, lattice networks, network capacity, queueing theory, routing, square grid, torus, uniform communication.

I. INTRODUCTION

LATTICE networks are widely used, for example, in distributed parallel computation [2], distributed control [3], satellite constellations [4], and wired circuits such as CMOS circuits and CCD-based devices [5]. Lattice networks are also known as grid [6] or mesh [7] networks. Moreover, the development of micro and nano-technologies [8] has also enabled the deployment of sensor networks for measuring and monitoring purposes [9]. 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.

We consider lattice networks, namely 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), and a similar lack of storage is typical in optical networks. In this paper, we focus on the analysis and design of routing algorithms that max-

imize 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 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 [10] and queueing approximations are used instead. Most of these approximations are based on Jackson's independence assumption [10]. The independence approximation works well under low rate but degrades rapidly as the rate increases. In this paper we propose a new analysis that captures the dependence between the queue distribution of different nodes and, therefore, achieves more accurate distributions for high rates. Using this model, we 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, central 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 [11]. In central 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 [12]. 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 this 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 with in-

Manuscript received July 28, 2004; revised June 15, 2005; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Srikant. This paper was presented in part at the Information Processing in Sensor Networks (IPSN) Conference, Berkeley, CA, Apr. 2004.

The authors are with the School of Computer Science and Communication, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne CH-1015, Switzerland (e-mail: guillermo.barrenetxea@epfl.ch; baltasar.berfull@epfl.ch; martin.vetterli@epfl.ch).

Digital Object Identifier 10.1109/TNET.2006.876187

finite buffers. We analyze capacity limits and provide optimal routing algorithms for both torus and square lattices. Then, in Section IV, we consider finite buffers and propose new approximate models to analyze the corresponding queueing network. Using these models, we compute the packet distribution in the queues and study the performance of routing algorithms with finite buffers. In Section V, we study the central data gathering scenario and, using the approximated models we proposed, we derive the optimal routing algorithm. In Section VI, we analyze the border data gathering problem.

A. Related Work

Capacity analysis of regular lattice network have already been addressed by diverse researchers [13]–[16]. Applying bisections methods [13], Leighton limited the maximum traffic that nodes can transmit in regular networks under a uniform traffic scenario. Using graph cut methods, Sun and Modiano [16] investigated the spare capacity, which is the capacity needed on each link to recover from a link or node failure, in an $N \times N$ torus topology under a uniform all-to-all traffic model. The same authors studied in [4] the performance of scheduling algorithms for packet transmission in node to node communications with a finite buffer space per node. They reported that when there is limited buffer at each node, the performances of different scheduling schemes are very similar and concluded that the buffer has a neutralizing effect on system throughput under different scheduling schemes. This work is related to [17], where the authors evaluate the throughput of two different packet scheduling policies for the infinite and finite buffer cases in a hypercube under the uniform traffic model. They observed that small buffer sizes can achieve throughput close to that of the infinite buffer case.

In our work, packet scheduling is fixed (FIFO) and the focus is on analyzing optimal routing algorithms that achieves the maximum throughput. Particularly, we show that the performance of different routing algorithms differs significantly when the buffers are limited. We also show that the required buffer to achieve a throughput close to that of the infinite buffer size reduces with the network size.

Routing in lattice networks has been thoroughly studied in the context of distributed parallel computation [2], where the system performance strongly depends on the routing algorithm. Various routing schemes have been studied through simulation by Maxemchuk [18].

Previous works that consider finite buffers are mainly based on Jackson's theorem [10]. Harchol-Balter and Black [19] considered the problem of determining the distribution on the queue sizes induced by the greedy routing algorithm in torus and square lattices. They reduced the problem into a product-form Jackson queue network and analyze it using standard queueing theory techniques. Mitzenmacher [20] 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. Unfortunately, the separation of the upper and lower bound, in the general case, grows as the square root of the number of nodes.

To reduce the complexity of the queueing network analysis, all the aforementioned papers make an independence approximation and consider the analysis of just one queue (node). In this

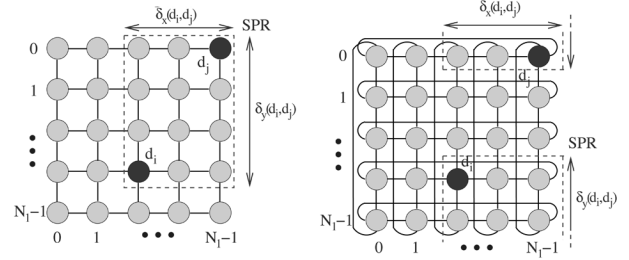


Fig. 1. Network model and least displacement of nodes $\{d_i, d_j\}$ in square (left) and torus (right) lattices. The shortest path region $\text{SPR}(d_i, d_j)$ Between nodes d_i and d_j is delimited in both cases by a dashed square.

paper we propose a new analysis that captures the dependence between the distributions on the queue size of different nodes. Using this analysis, we provide simple rules to design routing algorithms that maximize throughput for limited buffer nodes. These rules can be applied under different network topologies and different communications schemes.

Mergen and Tong [15] analyzed the effects of the MAC protocol on the network capacity, and more specifically, they studied the effect of multipacket reception capability on the capacity of wireless networks with regular structures under the uniform traffic model. For this scenario, they presented an optimal routing algorithms and a MAC scheme achieving the maximum capacity. For random and arbitrary wireless networks, Gupta and Kumar [21] studied the transport capacity and concluded that the total end-to-end capacity per node is $\mathcal{O}(1/\sqrt{n})$, where n is the number of nodes.

Neely, Rohrs and Modiano's [22], [23] 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 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 (Fig. 1). The subscripts “ s ” and “ t ” denote the square and the torus lattices, respectively. The square lattice is described by the graph $G_s(V, E_s)$ and the wrapped square or torus lattice by the graph $G_t(V, E_t)$. A torus lattice network is obtained from a square lattice by adding some supplementary links between opposite nodes located at the border of the lattice. 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 $d_i \in [0, N^2 - 1]$ 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 following a stationary Bernoulli distribution with a constant average rate of \mathcal{R} packets per node and per time slot. We denote by $T(d_i, d_j)$ the probability of d_i communicating to 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.

The network model is similar to the one discussed in [4]. We assume that time is slotted and a one-hop transmission consumes one time slot. 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* $\text{SPR}(d_i, d_j)$ 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, $\text{SPR}(d_i, d_j)$ in the square lattice is a rectangle with limiting corner vertices being d_i and d_j (Fig. 1). 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 $\delta_x(d_i, d_j) \in [-(N-1), N-1]$ and $\delta_y(d_i, d_j) \in [-(N-1), N-1]$ are the relative displacements (Fig. 1). 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).

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. Equivalently, we can consider that there are four queues per node, each one associated to one output link.

In the case of infinite buffers, a rate \mathcal{R} is said to be *achievable* if the total number of packets in the network stays bounded over time. Since the number of packets in a network with finite buffers is always bounded, a different definition for achievability is needed. Notice that, due to buffer overflow, there is a non-zero probability of packet loss at any rate. A rate \mathcal{R} is said to be *feasible* if the average loss probability is smaller than a given (sufficiently small) threshold τ_0 . Along this paper, we consider $\tau_0 = 5 \cdot 10^{-3}$.

Definition 1: An $N \times N$ network with infinite buffers is said to have capacity $C_{\{s,t\}}^{\{u, cg, bg\}}(N)$ if any rate $\mathcal{R} = C_{\{s,t\}}^{\{u, cg, bg\}}(N) - \epsilon$, $\forall \epsilon > 0$, is achievable. Moreover, any $\mathcal{R} \geq C_{\{s,t\}}^{\{u, cg, bg\}}(N)$ is not achievable.

The subscripts “s” and “t” indicate the network topology, square and torus lattices, respectively. The superscripts “u”, “cg”, and “bg” denotes the traffic model, that is, uniform, central data gathering, or border data gathering, respectively. The 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 $\text{SPR}(d_i, d_j)$. 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, for all $\{d_i, d_j, d_k, d_l\} \in V$,

$$\delta(d_i, d_j) = \delta(d_k, d_l) \rightarrow \Pi(d_i, d_j) = \Pi(d_k, d_l).$$

Definition 3: A routing algorithm Π is said to have maximum achievable rate $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ if any rate $\mathcal{R} = \mathcal{R}_{\text{sup}}^{\Pi}(N, Q) - \epsilon$, $\forall \epsilon > 0$, is feasible in a network of size $N \times N$ with buffer size Q using Π as routing algorithm. Moreover, any rate $\mathcal{R} \geq \mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ is not feasible.

Note that for $Q = \infty$, $\mathcal{R}_{\text{sup}}^{\Pi}(N, \infty)$ is the supremum of the set of achievable rates. Obviously, $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q) \leq C(N) \forall Q$.

III. UNIFORM COMMUNICATION 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:

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

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 on the network by proposing approximation models that allow us to simplify considerably the queueing network analysis.

A. Network Capacity

In the infinity buffer, the network capacity case can be easily derived applying bisection arguments [13], [14] to both torus and square lattices.

Lemma 1: The network capacity $C_{\{s,t\}}^u(N)$ for the uniform communication model is given by:

$$C_s^u(N) = \begin{cases} \frac{2\beta}{N} (1 - \frac{1}{N^2}), & \text{if } N \text{ is even,} \\ \frac{2\beta}{N}, & \text{if } N \text{ is odd,} \end{cases} \quad (2)$$

$$C_t^u(N) = \begin{cases} \frac{4\beta}{N} (1 - \frac{1}{N^2}), & \text{if } N \text{ is even,} \\ \frac{4\beta}{N}, & \text{if } N \text{ is odd,} \end{cases} \quad (3)$$

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

As we will see in next section, in the infinite buffer case, these upper bounds are actually tight and can be achieved by certain routing algorithms.

B. Optimal Routing Algorithms

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

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 $\lambda_{d_k}^{\Pi}$ the traffic arrival rate to node d_k according to Π :

$$\lambda_{d_k}^{\Pi} = \sum_{d_i \in V} \sum_{d_j \in V} T(d_i, d_j) F^{\Pi}(d_i, d_j, d_k). \quad (4)$$

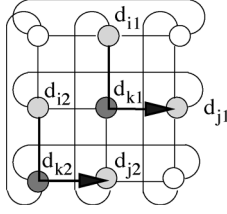


Fig. 2. The source-destination pair $\{d_{i1}, d_{j1}\}$ generates traffic that flows across node d_{k1} according to Π . If Π is space invariant, for any other node d_{k2} , we can find another source-destination pair $\{d_{i2}, d_{j2}\}$ with the same least displacement as $\{d_{i1}, d_{j1}\}$ that generates exactly the same traffic across d_{k2} as $\{d_{i1}, d_{j1}\}$ across d_{k1} .

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

$$\mathcal{R}_{\text{sup}}^{\Pi}(N, \infty) = C_t^u(N) \forall \Pi \in \{\text{shortest path space invariant}\}.$$

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. 2). 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 is constant:

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

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

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

Particularly, in a torus lattice under uniform traffic distribution

$$\bar{L}(N) = \begin{cases} \frac{N^3}{2(N^2-1)}, & \text{if } N \text{ is even} \\ \frac{1}{2}N, & \text{if } N \text{ is odd.} \end{cases} \quad (6)$$

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

$$\lambda = \frac{N^2 \mathcal{R} \bar{L}(N)}{N^2} = \mathcal{R} \bar{L}(N). \quad (7)$$

In the case of infinite buffers, the maximum achievable rate $\mathcal{R}_{\text{sup}}^{\Pi}(N, \infty)$ at which nodes can transmit while keeping the queues stable and the number of packets in the network bounded, is obtained by applying the stability condition in the nodes, that is

$$\lambda_{\text{sup}} = \mu_{\text{sup}} - \epsilon, \quad \forall \epsilon > 0 \quad (8)$$

where λ_{sup} is the maximum arrival rate per node and μ_{sup} is the maximum average number of packets that can be transmitted per unit of time. That is, $\mu_{\text{sup}} = 4$ for full-duplex communication channels and $\mu_{\text{sup}} = 2$ for half-duplex. Substituting (7) into (8), the maximum rate $\mathcal{R}_{\text{sup}}^{\Pi}(N, \infty)$ achieved by any space invariant routing algorithm Π is

$$\mathcal{R}_{\text{sup}}^{\Pi}(N, \infty) = \frac{\mu_{\text{sup}}}{\bar{L}(N)} = \frac{2\beta}{\bar{L}(N)}. \quad (9)$$

Combining (6) and (9), $\mathcal{R}_{\text{sup}}^{\Pi}(N, \infty)$ is equal to the network capacity given in (3). ■

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

Corollary 1: The network capacity $C_t^u(N)$ 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. 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.

Intuitively, the optimal 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 by enforcing a lower average traffic for these paths. This is the principle of *row-first (column-first)* routing [13]: 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. In the following, we prove that this simple algorithm is indeed optimal among all shortest path space invariant routing algorithms. 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, this is not a limiting restriction.

Theorem 2: For the square lattice 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

$$\lambda_{d_c}^{\Pi} \geq \mathcal{R}N. \quad (10)$$

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,

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

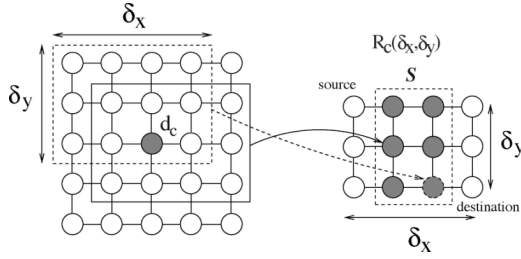


Fig. 3. For all source-destination pairs $\{d_i, d_j\}$ such that $\delta(d_i, d_j) = [3, 2]$, we obtain the set \mathcal{S} of relative positions of d_c in $R_c(\delta_x, \delta_y)$.

Given the symmetry of d_c , $\Gamma^\Pi(\delta_x, \delta_y, d_c)$ we have:

$$\Gamma^\Pi(\delta_x, \delta_y, d_c) = \Gamma^\Pi(\delta_y, \delta_x, d_c), \quad (11)$$

$$\Gamma^\Pi(\delta_x, \delta_y, d_c) = \Gamma^\Pi(|\delta_y|, |\delta_x|, d_c). \quad (12)$$

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

$$\lambda_{d_c}^\Pi = \sum_{\delta_x=-(N-1)}^{N-1} \sum_{\delta_y=-(N-1)}^{N-1} \Gamma^\Pi(\delta_x, \delta_y, d_c). \quad (13)$$

Using properties (12) and (11), we can reduce the analysis of $\Gamma^\Pi(\delta_x, \delta_y, d_c)$ to the cases $|\delta_x| \geq |\delta_y| \geq 0$. That is,

$$\begin{aligned} \lambda_{d_c}^\Pi = & 4 \sum_{\delta_x=0}^{N-1} \Gamma^\Pi(\delta_x, \delta_x, d_c) + 4 \sum_{\delta_x=0}^{N-1} \Gamma^\Pi(\delta_x, 0, d_c) \\ & + 8 \sum_{\delta_x=2}^{N-1} \sum_{\delta_y=1}^{\delta_x-1} \Gamma^\Pi(\delta_x, \delta_y, d_c). \end{aligned} \quad (14)$$

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 (14). 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. 3 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 $\Gamma^\Pi(\delta_x, \delta_y, d_c)$.

First, note that if $\delta_y > (N-1)/2$, 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 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 $\delta_y > (N-1)/2$ consists on using only the two most external paths of $\text{SPR}(d_i, d_j)$. Therefore, for any Π , $\Gamma^\Pi(\delta_x, \delta_y, d_c) \geq 0$ for all δ_x .

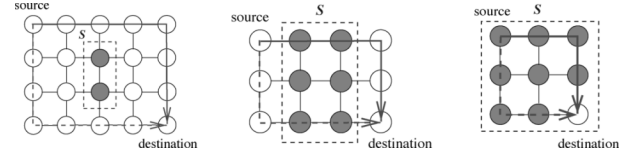


Fig. 4. $R_c(\delta_x, \delta_x)$ for three possible cases in a lattice network. Left: $\delta(i, j) = (4, 3)$; since $\delta_y > ((N-1)/2)$, \mathcal{S} does not fill completely any column of $\text{SPR}(d_i, d_j)$. Center: $\delta(i, j) = (3, 2)$; since $\delta_y \leq (N-1)/2$, \mathcal{S} fills $N - \delta_x$ columns. Right: $\delta(i, j) = (2, 2)$; \mathcal{S} fills all the δ_x columns. The arrows indicate two of the possible routing policies that generates the least possible traffic in \mathcal{S} .

If $\delta_y > (N-1)/2$ 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 $\Gamma^\Pi(\delta_x, \delta_y, d_c) \geq (\mathcal{R}/(N^2 - 1))(N - \delta_x)$. Note that there are many routing policies that achieve this lower bound, as for instance, the routing algorithm that uses only the two most external paths.

If $\delta_y \leq (N-1)/2$, \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} . That is, $\Gamma^\Pi(\delta_x, \delta_y, d_c) = (\mathcal{R}/(N^2 - 1))(\delta_x + \delta_y)$. 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. These three cases are illustrated in Fig. 4.

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

$$\Gamma^\Pi(\delta_x, \delta_y, d_c) \geq \begin{cases} \frac{\mathcal{R}}{N^2-1}(\delta_x + \delta_y), & \delta_x \leq \frac{N-1}{2}, \delta_y \leq \frac{N-1}{2} \\ \frac{\mathcal{R}}{N^2-1}(N - \delta_x), & \delta_x > \frac{N-1}{2}, \delta_y \leq \frac{N-1}{2} \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

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

Using (15) and (14), the total traffic that flows through d_c is bounded as

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

and evaluating summations yields to $\lambda_{d_c}^\Pi \geq \mathcal{R}N$. ■

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 that a network capacity-achieving routing algorithm Π generates a traffic distribution where there exists a node d_x such that $\lambda_{d_x}^\Pi > \lambda_{d_c}^\Pi$. Then, from Theorem 2, $\lambda_{d_x}^\Pi > \mathcal{RN}$. Particularly, we have that $(\lambda_{d_x}^\Pi)_{\text{sup}} > \mathcal{R}_{\text{sup}}^\Pi(N, \infty)N$.

The maximum rate per node $\mathcal{R}_{\text{sup}}^\Pi(N, \infty)$ achieved by Π is obtained by imposing stability conditions for d_x , that is

$$(\lambda_{d_x}^\Pi)_{\text{sup}} = \mu_{\text{sup}} - \epsilon, \quad \forall \epsilon > 0$$

and therefore,

$$\mathcal{R}_{\text{sup}}^\Pi(N, \infty) < \frac{(\lambda_{d_c}^\Pi)_{\text{sup}}}{N} = \frac{\mu_{\text{sup}} - \epsilon}{N}.$$

Using the capacity formula of a square lattice (2) we have

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

and consequently, Π does not reach capacity. ■

The factor that limits the maximum achievable rate in the network is the amount of traffic routed through the center node d_c . 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, that is, row-first (column-first) routing.

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

Proof: From Theorem 2, we know that $\lambda_{d_c}^{\text{r-f}} = \mathcal{RN}$. Moreover, the traffic distribution induced by row-first routing is such that $\lambda_{d_c}^{\text{r-f}} > \lambda_{d_x}^{\text{r-f}}$ for all $d_x \in V/d_c$.

In the case of infinite buffers, the maximum achievable rate $\mathcal{R}_{\text{sup}}^{\text{r-f}}(N, \infty)$ at which nodes can transmit information while keeping the number of packets in the network bounded, is obtained by applying the stability condition in the most loaded node, that is, d_c :

$$(\lambda_{d_c}^{\text{r-f}})_{\text{sup}} = \mu_{\text{sup}} - \epsilon, \quad \forall \epsilon > 0. \quad (16)$$

Using Theorem 2 and recalling the capacity formula (2), we obtain: $\mathcal{R}_{\text{sup}}^{\text{r-f}}(N, \infty) = \mu_{\text{sup}}/N = C_s^u(N)$. ■

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

IV. UNIFORM COMMUNICATION WITH FINITE BUFFERS

When finite buffers are considered, the maximum rate per node is clearly reduced due to buffer overflow. Overflow losses will first appear in the most loaded node, which determines the maximum achievable rate $\mathcal{R}_{\text{sup}}^{\text{r-f}}(N, \infty)$.

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 average traffic (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 .

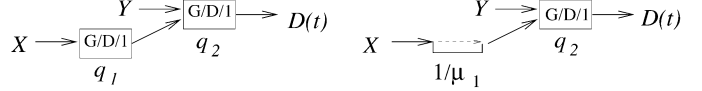


Fig. 5. The total number of packets in a two-queue system remains the same if the first stage queue is replaced by a pure delay of T time slots.

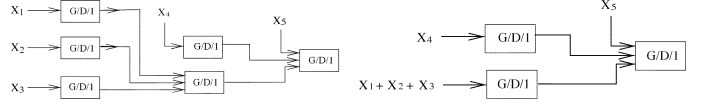


Fig. 6. The number of packets in the head node of the tree network (left) is the same as in the two-stage equivalent model (right).

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

Computing the network capacity for different buffer sizes Q requires analyzing the associated 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 even for the simplest cases [10]. In this section, we introduce some approximations that simplify this analysis and provide meaningful theoretical results that, as experimentally shown later, are close to the results obtained by simulation.

First, we 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 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 clearly identical. Moreover, due to the independence of packet generation, we assume that these arrival distributions are also independent. Therefore, we approximate the distribution on the queue size at d_m as the addition of these four i.i.d. distributions and compute it as the convolution of each individual queue. This way, we reduce the problem to computing the distribution on the size of only one of these i.i.d. queues, q_m , associated with the output link l_m in d_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:* If l_m is a full-duplex channel, 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 [22], [23] on equivalent models for tree networks of deterministic service time queues. We begin by reviewing the main results in [22], [23], and then we show how these results can be applied to our problem.

Theorem 3: [22] The total number of packets in a twoqueue 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: [23] The analysis of the queue distribution in the head node of a multi-stage tree system can be reduced to the analysis of a 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.

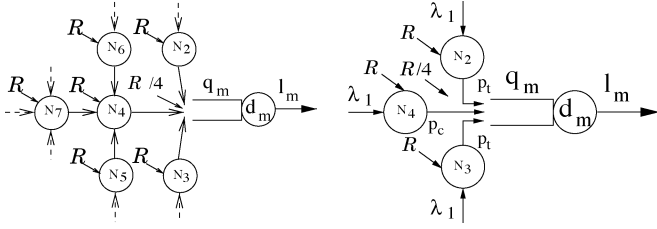


Fig. 7. Tree network and its equivalent two-stage model. Left: tree network associated to one output link l_m of d_m . Right: its two-stage equivalent model.

Fig. 5 shows the equivalence provided by Theorem 3. Fig. 6 shows a tree system and its two-stage equivalent model. Importantly, these equivalences do not require any assumption about the nature of the exogenous traffic. The only necessary condition is that all queues of the tree network have a deterministic service time T , and the exogenous 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 of all the nodes sending traffic through l_m . 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. 7), where we only consider the three neighbors located one hop away from d_m preserving the traffic generated by the entire network that flows through l_m .

Note that even if the two-stage model of Theorem 4 is an *equivalent* model for tree networks, the reduced two-stage model of Fig. 7 is an *approximation* model. The reason is that the network associated to q_m is not exactly a tree and, therefore, it does not correspond exactly to the network of Theorem 4 (Fig. 6): in addition to the exogenous inputs generated at each node, there is also traffic leaving the network at each node that corresponds to the traffic that has reached its destination or the traffic that does not travel through q_m . Under the uniform communication model, the average traffic that reaches its destination and leave the network is equal to \mathcal{R} at all nodes. However, as the network size increases, \mathcal{R} decreases as $\mathcal{O}(1/N)$, and consequently, the departing traffic at each node becomes negligible compared to the traffic that flows through the same node. Hence, the network of queues associated to q_m is *approximately* a tree network, and the two-stage model provides an approximated network, which is more precise as the network size increases.

According to Theorem 4, 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.

Note that 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

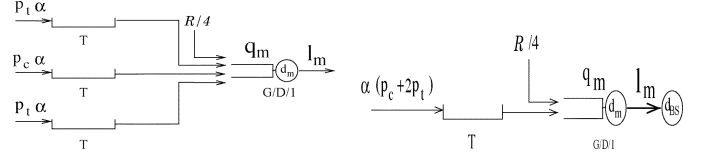


Fig. 8. Two-stage equivalent networks: if we replace the queues of the first stage by pure delays of T time slots (left), the total number of packets in the approximated model remains constant. In terms of number of packets, this is equivalent to injecting all the arrivals to a single pure delay (right).

column, and p_t the probability of turning. These probabilities are easy to calculate at d_m :

$$p_c = \begin{cases} \frac{N-1}{N+1}, & \text{for the square lattice,} \\ \frac{N/2-1}{N/2+1}, & \text{for the torus,} \end{cases} \quad (17)$$

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

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 as l_m .

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:

$$\lambda_{q_m} = \mathcal{R}/4 + \lambda_1(p_c + 2p_t) \quad (19)$$

where λ_1 is the total arrival rate to the neighbors of d_m (Fig. 7). For row-first routing, λ_1 is equal to

$$\lambda_1 = \begin{cases} \frac{\mathcal{R}N}{4}, & \text{for the square lattice,} \\ \frac{\mathcal{R}N}{8}, & \text{for the torus.} \end{cases} \quad (20)$$

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, recalling the capacity formula, $\lambda_1 = \alpha$ for both torus and square lattices. Putting everything together, the resulting approximation model is shown in Fig. 9.

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 exogenous traffic distribution as long as it is stationary and independent among the different sources. Using now these reduced approximation models, we can derive some interesting results about the buffer requirements to achieve a certain relative capacity.

Theorem 5: The buffer size Q required to achieve a certain relative capacity α decreases with the network size N .

Proof: By Theorem 3, the total number of packets in the two-stage approximated model is the same as a system where the queues of the first stage have been replaced by pure delays of T time slots, and this is equivalent to injecting all the arrivals into a single pure delay (Fig. 8). The total average arrival rate λ_{S_1} to the first queues is

$$\lambda_{S_1} = \alpha(p_c + 2p_t) = \alpha(N-1)/N. \quad (21)$$

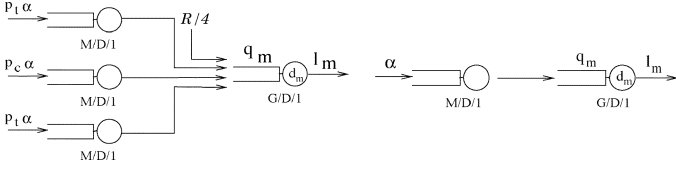


Fig. 9. Approximation models. Left: two-stage model used to analyze the distribution on the size of q_m . Right: two-queue model, a model without crossing packets and neglecting exogenous traffic in d_m .

Therefore, note that for a fixed α and large N , the total number of packets in this two queue model is almost constant with N .

We can decompose the total number of packets $S(t)$ in the two-stage approximated model 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, $S(t) = S_1(t) + S_h(t)$. 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. 9, where no buffer is needed in the head node. ■

Subsequently, we can simplify our model even further while still keeping the important properties that determine the queue size distribution. Note that, since p_t is $\mathcal{O}(1/N)$, we can assume that for large networks the number of packets turning at d_m is negligible, i.e., the packets arrive at q_m only from the neighbor located in the same row or column as l_m . Similarly, the exogenous traffic generated at d_m , goes also asymptotically to zero ($\mathcal{O}(1/N)$) as 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 arrival rate equal to α (Fig. 9). 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 independent and identically distributed queues associate to the four outgoing links from d_m :

$$P_m(k) = \begin{cases} \binom{4}{k} (1 - \alpha)^{4-k} \alpha^k, & \text{for } 0 \leq k \leq 4 \\ 0, & \text{otherwise.} \end{cases} \quad (22)$$

Note that both approximation models proposed (Fig. 9) are asymptotically exact.

Fig. 10 shows the distribution on the size of q_m obtained by simulating the whole queueing network, the two-stage model, the two-queue model, and the usual M/D/1 approximation for different values of α in a 121×121 square lattice.

For the M/D/1 approximation, we simply apply Jackson's Theorem and consider that each queue in the network is M/D/1 independent of other queues [20]. Therefore, we approximate q_m by a M/D/1 queue with a Poisson arrival with rate α .

Both the two-stage and two-queue models allow very good analysis in low and medium load. 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. For instance, to approximate the network by a tree, we neglected

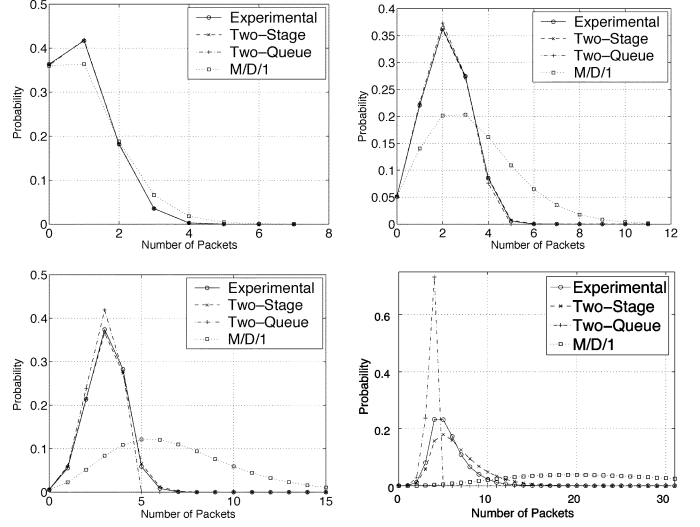


Fig. 10. Distribution on the queue size at d_m for values of $\alpha = [0.225, 0.525, 0.725, 0.925]$ (from top-left to bottom-right) in a 121×121 square lattice with full-duplex links.

the traffic leaving the network at each (destination) node, which increases with \mathcal{R} .

Both approximation models that we propose clearly outperform, at all rates, the M/D/1 model. The reason is that the M/D/1 approximation neglects the existing correlation between the traffic and queue occupancy of neighbor nodes, while a simple two-stage model captures this correlation. Therefore, the M/D/1 model only approximates the distribution closely under low load conditions, while the independence approximation is valid. Under medium and high loads, the independence assumption does not hold, and the approximation quality degrades rapidly. For $\alpha = 0.525$, we already observe that this approximation is far from the distribution obtained experimentally.

Fig. 14 shows the distribution on the queue size at d_m , for a constant relative capacity $\alpha = 0.75$, as a function of the network size N . As expected, both approximations become closer to the experimental distribution as N increases and the packet distribution converges to the two-queue model (Fig. 9). 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 N .

2) Half-Duplex Communication Channels: For half-duplex, 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. 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. 11). 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

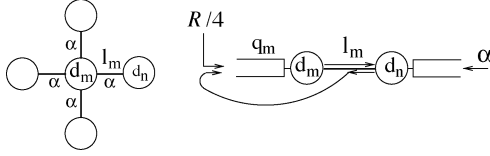


Fig. 11. Half-duplex approximation model. Left: network with half-duplex links. Right: approximated model for the queue associated to l_m .

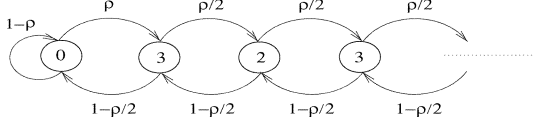


Fig. 12. Markov chain model for the half-duplex links model.

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. 11.

We denote by ρ_m the utilization factor of q_m . That is, $\rho_m = \lambda_{q_m} / \mu_{q_m} = \alpha / \mu_{q_m}$, 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 :

$$p_s = \frac{1}{2} + \frac{1}{2}(1 - \rho_n) = 1 - \rho_n/2; \quad (23)$$

We model q_m service time as a geometric distribution with parameter p_s : if d_m does not capture l_m this time slot, the probability to capture it the next time slot remains the same. As in the full-duplex case, we approximate arrivals to d_n as a Poisson distribution with parameter α . Accordingly, inter-arrival 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 , $\{X_m(t) | t > 0\}$ 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

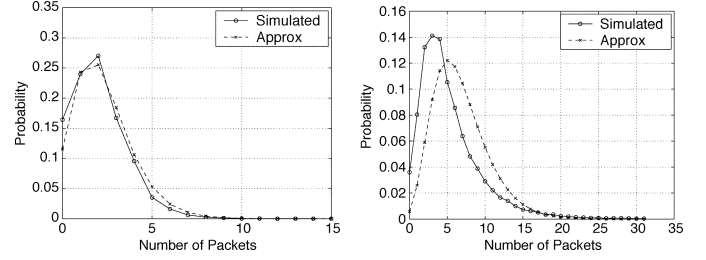


Fig. 13. Distribution on the queue size at d_m for $\alpha = 0.725$ (left) and $\alpha = 0.925$ (right) in a 121×121 square lattice with half-duplex links.

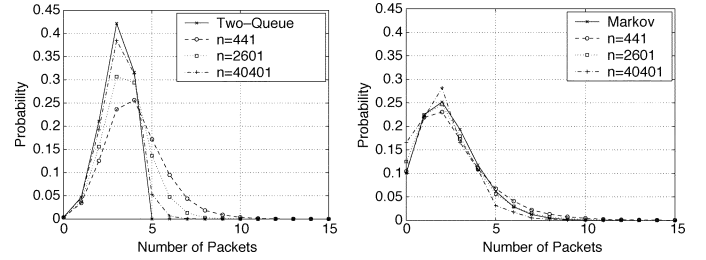


Fig. 14. Distribution on the queue size at d_m for $\alpha = 0.75$ and different network sizes with full-duplex (left) and half-duplex (right) links.

probability matrix $P_m(j, k)$ associated to $\{X_m(t) | t > 0\}$ can be approximated by

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

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

whose transition graph is shown in Fig. 12.

Fig. 13 shows the distribution on the queue size at d_m for different values of α in a 121×121 square lattice 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 full-duplex case, beyond this traffic intensity, some assumptions are no more valid and the approximation quality degrades.

Fig. 14 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. CENTRAL DATA GATHERING MODEL

Using a similar methodology as before, we analyze now the central data gathering model. Under this model, every node transmits information to a particular and previously designated node d_{BS} denoted *base station* that can be located anywhere

in the network. Note that in this case, routing in a torus is a particular case of routing in a square lattice since 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. We bound the network capacity $C_s^{cg}(N)$ and analyze optimal routing algorithms for infinite buffers, and then, we analyze the effect of finite buffers.

A. Network Capacity

Lemma 2: The network capacity $C_s^{cg}(N)$ for the central data gathering communication model in a square lattice is upper bounded as follows:

$$C_s^{cg}(N) \leq \frac{|\varphi(d_{BS})|}{N^2 - 1}. \quad (24)$$

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 [13] to these links yields the result. ■

Since in the links that limit the capacity the information flows only in one direction (from the nodes to d_{BS}), 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 . In the infinity buffer case, capacity achieving routing algorithms are characterized by the following Lemma:

Lemma 3: 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

$$\lambda_l^\Pi = \frac{\mathcal{R}(N^2 - 1)}{|\varphi(d_{BS})|} \quad \text{for all } l \in \varphi(d_{BS}). \quad (25)$$

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

$$\sum_{l \in \varphi(d_{BS})} \lambda_l^\Pi = \mathcal{R}(N^2 - 1). \quad (26)$$

In central data gathering, the most loaded link in the network obviously belongs to $\varphi(d_{BS})$. Applying the stability condition for these unitary capacity links

$$\max_{l \in \varphi(d_{BS})} \lambda_l^\Pi = 1 - \epsilon, \quad \forall \epsilon > 0. \quad (27)$$

Combining (26) and (27), the result follows. ■

Corollary 5: The network capacity $C_s^{cg}(N)$ of a square lattice for central data gathering model is equal to the upper bound given by (24).

C. Routing With Finite Buffers

Lemma 3 establishes that the only necessary and sufficient condition for a routing algorithm to achieve capacity with infinite buffers 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.

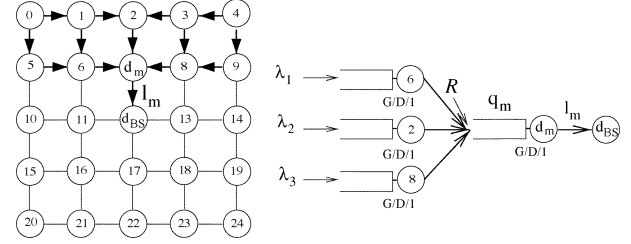


Fig. 15. Two-stage equivalent model. Left: tree network where l_m is the head node. Right: its two-stage equivalent model.

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 in this case, 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 for infinite buffers. For instance, a simple routing algorithm that achieves capacity is the *random greedy algorithm* [13], where nodes use row-first or column-first as routing algorithm with equal probability.

To analyze the network capacity 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 this tree to its two-stage equivalent model and obtain the packet distribution in d_m by analyzing 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 in the infinite buffer case.

The bottleneck of the network is clearly located in the four neighbors of d_{BS} . Moreover, if Π achieves capacity when buffers are infinite, the total arrival traffic to d_{BS} according to Lemma 3 is uniformly distributed among the four links in $\varphi(d_{BS})$. Due to the independence of packet generation, we can assume that 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, say 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. 15).

We consider now only those nodes in the network 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. 15). Note that in this case there is not traffic leaving the network at each node as in the uniform communication case. Therefore, the two-stage model is not an approximation but an exact model for all rates.

The arrivals to the three nodes of the first stage are the addition of all the traffic generated by the network that goes through l_m . If Π achieves capacity for infinite buffers, by Lemma 3, the total average traffic that flows through l_m is

$$\lambda_{l_m} = \frac{\mathcal{R}(N^2 - 1)}{4}. \quad (28)$$

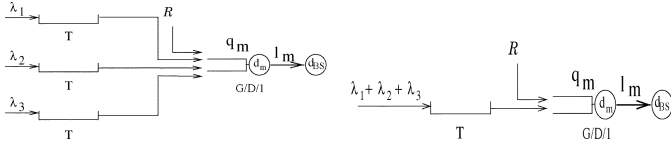


Fig. 16. Two-stage equivalent model. Two-stage model where the first stage queues has been replaced by pure delays of T time slots (left). This is equivalent to injecting all the arrivals to a single pure delay (right).

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. 15). These three nodes have to route all the traffic that goes through l_m except the traffic generated by d_m itself. That is,

$$\lambda_1 + \lambda_2 + \lambda_3 = \frac{\mathcal{R}(N^2 - 1)}{4} - \mathcal{R}. \quad (29)$$

We obtain the distribution on the size of q_m by analyzing the distribution at the head node of the two-stage model, which obviously depends on the values of λ_1 , λ_2 and λ_3 . Particularly, we are interested in finding the routing algorithm Π that, for a given Q , achieves the maximum $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$. This is equivalent to minimizing the number of packets in q_m for any given \mathcal{R} . 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 4: 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 , $i \in \{1, 2, 3\}$ 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, $\lambda_i = \lambda_t$ for $i = 1, 2$, or 3 , and 0 otherwise.

Proof: By Theorem 3, the total number of packets in the two-stage model (Fig. 15) is the same as a system where the first stage queues has been replaced by pure delays of T time units. In terms of number of packets in the system, this is equivalent to injecting all the arrivals to a single pure delay (Fig. 16). 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. ■

Using the two-stage equivalent model and Lemma 4, we can easily design routing algorithms that minimize packet overflow. Consider a routing algorithm Π that achieves capacity for the infinite buffer case. That is, Π generates the optimal traffic distribution to the neighbor nodes of d_{BS} according to Lemma 3. As we have previously pointed out, maximizing $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ is equivalent to minimizing the number of packets in the most loaded nodes. Therefore, if we want Π to be also optimal for finite buffers, it needs to generate the optimal arrival distribution (as given in Lemma 4) in all the nodes.

Intuitively, if Π achieves the maximum $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$, the input traffic to the most loaded node d_m has to arrive from only one of its neighbors, say d_t . This way, we minimize the

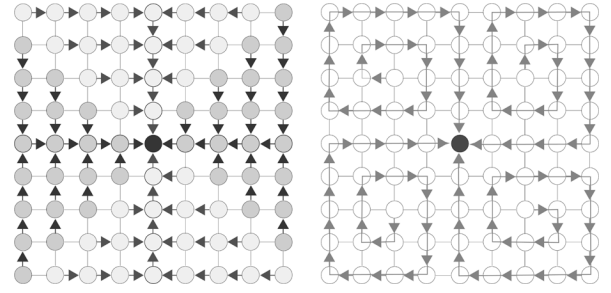


Fig. 17. Routing algorithms for central data gathering with finite buffers. Left: cross routing. Right: snail routing.

overflow losses in d_m . However, by routing all the traffic that arrives at d_m through d_t , the congestion problem is translated to d_t . Consequently, d_t is now the most loaded node, and we can apply the same argument as before. Furthermore, as the network size increases, the difference between the traffic that flows through d_m and d_t goes asymptotically to zero.

It is clear now that to obtain the routing algorithm that minimizes the number of packets in all nodes, and equivalently, minimizes the overflow losses, Lemma 4 has to be applied to all nodes. That is, the optimal routing algorithm is such that all nodes receive traffic from only one neighbor. We apply now this general principle to design algorithms that minimize overflow losses.

The shortest path routing algorithm that implements this principle consists in the following. In the $N \times N$ square lattice, there are $2(N - 1)$ nodes that have only 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 in 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 lattice. Diagonal nodes forward packets towards only 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 forwarding node. We denote this routing algorithm as *cross routing* (Fig. 17).

Cross routing generates the optimal node arrival distribution among all shortest path routing algorithms. Although the most loaded nodes, that is, nodes located close to d_{BS} , receive packets from more than one neighbor, most of the traffic arrives mainly from one. The average arrival rates generated in d_m by cross routing 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.

However, according to Lemma 4, 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 this condition recursively, the set of optimal routing algorithms is such that it divides the network into four disjoint subsets of $(N^2 - 1)/4$ nodes and joins them with a single path that does not pass twice through the same node and ends in d_{BS} . We denote the optimal routing algorithms set as traveling salesman (TS) routing.

Fig. 17 shows an example of a TS routing algorithm where the traffic flows toward d_{BS} following a spiral. We denote this routing algorithm as *snail routing*. Clearly, Snail routing belongs to TS routing and, according to Lemma 4, generates the optimal arrival distribution in all nodes.

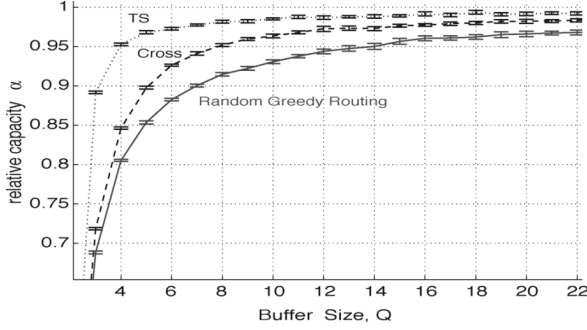


Fig. 18. Routing for central data gathering: maximum relative capacity α achieved by different routing algorithms in a 21×21 square lattice for different buffer sizes Q with the 95% confidence intervals.

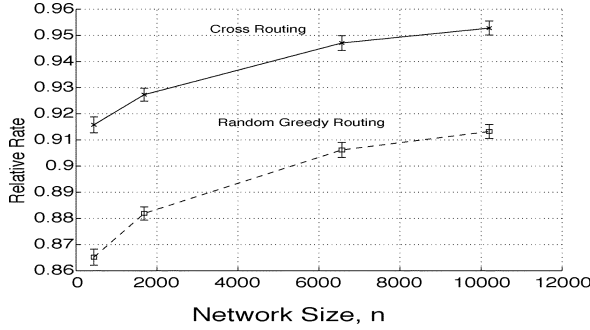


Fig. 19. Routing for central data gathering: maximum rate achieved by cross routing and greedy routing relative to snail routing for a fixed buffer size $Q = 5$ for different network sizes n with the 95% confidence intervals.

Note that Lemma 4 is valid for any arrival distribution, and consequently, it holds for any exogenous traffic generation process at the nodes. Thus, even if the maximum achievable rate $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ clearly depends on the traffic generation process, cross routing and TS routing still achieve the highest maximum rate among the shortest path and non-shortest path routing algorithms, respectively.

Fig. 18 shows the maximum relative capacity $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)/C_{\text{cg}}(N)$ achieved by random greedy routing, cross routing, and TS routing in a 21×21 square lattice as a function of the buffer size Q .

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

Fig. 19 shows the maximum rate achieved by different routing algorithms relative to the maximum rate achieved by the snail routing for $Q = 5$ as a function of the network size N . Since all routing algorithms analyzed are asymptotically optimal with the network size, the performance gap between snail routing and these algorithms decreases as N increases.

Although TS routing achieves the maximum $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ 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 \bar{L}_{TS} for any TS routing algorithm is $\mathcal{O}(N^2)$, while for any shortest path routing, \bar{L}_{s-p} is $\mathcal{O}(N)$.

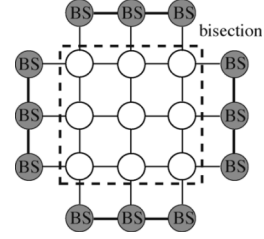


Fig. 20. Border data gathering model.

TS routing represents an extreme case of the existing tradeoff between $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ and delay, achieving the optimal rate per node drastically increases the delay. Equivalently, since most of the energy is commonly consumed in the transmission process, to increase the average path length is equivalent to increase the average power consumption in the network.

VI. BORDER DATA GATHERING MODEL

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. 20). 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 in the infinite buffer case based only on flow arguments. Then, we present the set of routing algorithms that achieve capacity for infinite buffers and 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

Lemma 5: The network capacity $C_s^{bg}(N)$ for the border data gathering communication model in a square lattice is upper bounded as follows:

$$C_s^{bg}(N) \leq \frac{4}{N-2}. \quad (30)$$

The network capacity is determined by the links connecting inner nodes to base stations (Fig. 20). Since the information flows through these links only in one direction, 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 5 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 are characterized by the following Lemma:

Lemma 6: 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,

$$\lambda_l^{\Pi} = \frac{\mathcal{R}(N-2)}{4}, \quad \text{for all } l \in \varphi(S_{BS}). \quad (31)$$

The proof of this lemma is similar to the proof of Lemma 3.

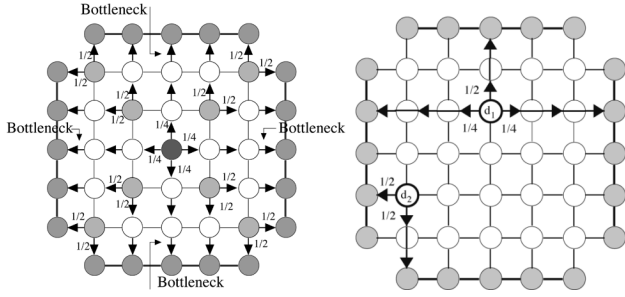


Fig. 21. Routing in border data gathering. Left: optimal shortest path routing. Right: uniform data gathering. Packets generated in a node are routed with equal probability to the two closest base stations located in the same row/column as the node (d_2). If there are two base stations in the same column/row at the same distance (d_1), both have equal probability.

First 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. 21. The maximum rate $\mathcal{R}_{sup}^{s-p}(N, \infty)$ 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. 21):

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

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

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 and column as the node. If there are more than one base stations in the same row or column at the same distance, it chooses any of them with equal probability. We denote this routing algorithm as *uniform data gathering* (Fig. 21). Uniform data gathering satisfies the capacity condition (31) and therefore, $\mathcal{R}_{sup}^{uniform}(N, \infty) = C_s^{bg}(N)$.

C. Routing With Finite Buffers

To derive the routing algorithm that achieves the maximum $\mathcal{R}_{sup}^{\Pi}(N, Q)$ for finite buffers, we apply the condition we derived in Section V-B: the optimal routing policy is such that nodes receive 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, no routing algorithm reaches capacity and behaves optimally for finite buffers.

Lemma 7: The optimal queue condition for finite buffers that minimizes the number of packets in the queues, and the capacity condition (31) cannot be both satisfied.

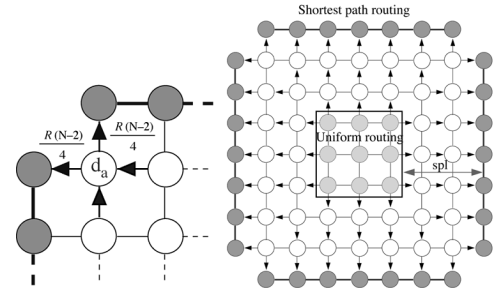


Fig. 22. Maximum rate trade-off in border data gathering: Left: the nodes located close to the edges carry more traffic. Right: adaptive routing.

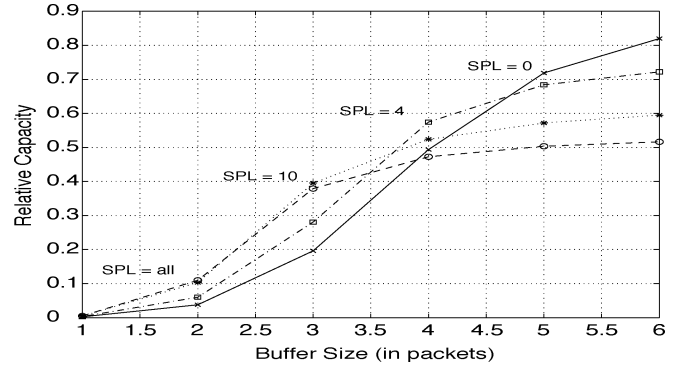


Fig. 23. Adaptive routing: maximum relative capacity achieved by adaptive routing with different shortest-path limit values as a function of the buffer size in a 41×41 square lattice.

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. 22. Notice that it is enough to focus ourselves on the traffic that goes through a certain node. By Lemma 6, d_a has to carry the traffic of at least two links in $\varphi(S_{BS})$, that is, $\lambda_{d_a} \geq 2(\mathcal{R}(N-2)/4)$. That is, d_a receives traffic from more than one neighbor. Otherwise, there is a link l such that $\lambda_l > (\mathcal{R}(N-2)/4)$, and Π does not achieve capacity. On the other hand, if d_a receives traffic from only one neighbor, λ_{d_a} is upper bounded as $\lambda_{d_a} \leq (\mathcal{R}(N-2)/4)$, and (31) is not satisfied. ■

Lemma 7 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 buffers. Therefore, we propose an adaptive routing algorithm that depends on the buffer size Q . 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* (Fig. 22). When spl is equal to zero, adaptive routing is equivalent to uniform data gathering, and the more loaded nodes receive 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.

Fig. 23 shows the values of $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ achieved by adaptive routing algorithm with different values of spl in a 41×41 square lattice, as a function of the buffer size Q . First, note the trade-off between the rate $\mathcal{R}_{\text{sup}}^{\Pi}(N, Q)$ achieved for big and small buffer sizes: no routing strategy can achieve high rates for both extremes. For high values of Q , the optimal routing strategy consists on choosing $\text{spl} = 0$. As Q 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.

REFERENCES

- [1] G. Barrenetxea, B. Beferull-Lozano, and M. Vetterli, "Lattice sensor networks: Capacity limits, optimal routing and robustness to failures," presented at the Information Processing in Sensor Networks (IPSN) Conf. Berkeley, CA, Apr. 2004.
- [2] R. Duncan, "A survey of parallel computer architectures," *IEEE Computer*, vol. 23, no. 2, pp. 5–16, Feb. 1990.
- [3] R. D'Andrea and G. E. Dullerud, "Distributed control design for spatially interconnected systems," *IEEE Trans. Autom. Contr.*, vol. 48, no. 9, pp. 1478–1495, Sep. 2003.
- [4] J. Sun and E. Modiano, "Routing strategies for maximizing throughput in satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 2, pp. 273–286, Feb. 2004.
- [5] A. El Gamal, "Trends in CMOS image sensor technology and design," in *IEEE Int. Electron Devices Meeting Dig.*, Dec. 2002, pp. 805–808.
- [6] D. Bhatia, T. Leighton, F. Makedon, and C. H. Norton, "Improved algorithms for routing on two-dimensional grids," in *Proc. 18th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG'92)*, 1992, Lecture Notes in Computer Science vol. 657, pp. 114–122.
- [7] J. F. Sibeyn, Overview of Mesh Results. Max-Planck-Institut für Informatik, Saarbrücken, Germany, Tech. Rep. MPI-I-95-1-018, 1995.
- [8] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for "smart dust"," in *Proc. 5th Annu. ACM/IEEE Int. Conf. Mobile Computing and Networking (MobiCom'99)*, Aug. 1999, pp. 271–278.
- [9] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
- [10] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- [11] M. D. Grammatikakis, D. F. Hsu, M. Kraetzl, and J. F. Sibeyn, "Packet routing in fixed-connection networks: a survey," *J. Parallel Distrib. Comput.*, vol. 54, no. 2, pp. 77–132, Nov. 1, 1998.
- [12] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, Feb. 2003.
- [13] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures*. San Mateo, CA: Morgan-Kaufman, 1991.
- [14] E. Modiano and A. Ephremides, "Efficient algorithms for performing packet broadcasts in a mesh network," *IEEE/ACM Trans. Netw.*, vol. 4, no. 4, pp. 639–648, Aug. 1996.
- [15] G. Mergen and L. Tong, "Capacity of regular ad hoc networks with multipacket reception," in *Proc. Allerton Conf. Commun., Control and Computing*, 2002.
- [16] J. Sun and E. Modiano, "Capacity provisioning and failure recovery for low earth orbit satellite constellation," *Int. J. Satellite Commun. Netw.*, vol. 21, pp. 259–284, Jun. 2003.
- [17] E. Varvarigos and D. P. Bertsekas, "Performance of hypercube routing schemes with or without buffering," *IEEE/ICM Trans. Netw.*, vol. 2, no. 9, pp. 299–311, Jun. 1994.
- [18] N. F. Maxemchuk, "Comparison of deflection and store-and-forward techniques in the Manhattan Street and Shuffle-Exchange Networks," in *IEEE INFOCOM*, 1989, vol. 3, pp. 800–809.
- [19] M. Harchol-Balter and P. Black, "Queueing analysis of oblivious packet-routing networks," in *Proc. 5th Annu. ACM/SIAM Symp. Discrete Algorithms*, Arlington, VA, Jan. 1994, pp. 583–592.
- [20] M. Mitzenmacher, "Bounds on the greedy routing algorithm for array networks," in *Proc. 6th Annu. Symp. Parallel Algorithms and Architectures*, New York, Jun. 1994, pp. 346–353.
- [21] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
- [22] M. J. Neely, C. E. Rohrs, and E. Modiano, "Equivalent models and analysis for multistage tree networks of deterministic service time queues," in *Proc. 38th Annu. Allerton Conf. Communication, Control and Computing*, Oct. 2000.
- [23] M. J. Neely, C. E. Rohrs, and E. Modiano, "Equivalent Models for Analysis of Deterministic Service Time Tree Networks. MIT-LIDS, Tech. Rep. P-2540, Mar. 2002.



Guillermo Barrenetxea (S'05) received the B.Sc. and the M.Sc. degrees in telecommunication engineering from Public University of Navarre (Universidad Publica de Navarra), Spain, in 2000. He received the Ph.D. degree in communication systems from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2005.

From September to May 2000, he was a Research Assistant at Institut Eurecom, Sophia-Antipolis, France. From June to October 2000, he was an intern at British Telecom (BT), Adastral Park, Ipswich, U.K. His research interests include network routing, sensor networks, and source coding.



Baltasar Beferull-Lozano (S'01–M'02) was born in Valencia, Spain, in 1972. He received the M.Sc. degree in physics from Universidad de Valencia (UV), Spain, in 1995 (with First in Class Honors) and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Southern California (USC), Los Angeles, in 1999 and 2002, respectively. His Ph.D. work was supported by a National Graduate Doctoral Fellowship from the Ministry of Education of Spain.

From January 1996 to August 1997, he was a Research Fellow Assistant at the Department of Electronics and Computer Science, Universidad de Valencia, and from September 1997 to September 2002, he was a Research Fellow Assistant in the Department of Electrical Engineering, the NSF Research Integrated Media Systems Center and the Signal and Image Processing Institute (SIPI), at USC. He has also worked for AT&T Shannon Laboratories. From October 2002 to June 2005, he was a Research Associate in the Department of Communication Systems at the Swiss Federal Institute of Technology (EPFL), Lausanne, and a Senior Researcher within the Swiss National Competence Center in Research on Mobile Information and Communication Systems. From July 2005 to November 2005, he was a Visiting Professor at Universidad de Valencia and Universidad Politécnica de Valencia. Since December 2005, he has been a Research Professor with the Instituto de Robótica, Escuela Técnica Superior de Ingeniería at Universidad de Valencia. He has served as a member of the Technical Program Committees for several ACM and IEEE international conferences. His research interests are in the general areas of signal and image processing, distributed signal processing and communications for wireless networks, information theory, and communication theory.

At USC, Dr. Beferull-Lozano received several awards including the Best Ph.D. Thesis paper Award in April 2002 and the Outstanding Academic Achievement Awards in April 1999 and April 2002.



Martin Vetterli (F'95) received the Dipl. El.-Ing. degree from ETH Zurich (ETHZ), Switzerland, in 1981, the M.S. degree from Stanford University, Stanford, CA, in 1982, and the Dr. Sci. degree from EPF Lausanne (EPFL), Switzerland, in 1986.

In 1986, he joined Columbia University in New York, first with the Center for Telecommunications Research and then with the Department of Electrical Engineering where he was an Associate Professor. In 1993, he joined the University of California at Berkeley, where he was Full Professor until 1997.

Since 1995, he has been a Professor at EPFL, where he headed the Communication Systems Division in 1996/1997 and now heads the Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research on mobile information and communication systems. He has also been Vice-President for International Affairs at EPFL since October 2004. He has held visiting positions at ETHZ (1990) and Stanford (1998). He is the co-author of a textbook, *Wavelets and Subband Coding* (Prentice-Hall, 1995), and of over 100 journal papers. His research interests are in the areas of applied mathematics, signal processing and communications.