# Efficient routing with small buffers in dense networks

Guillermo Barrenetxea*, Baltasar Beferull-Lozano* and Martin Vetterli*†
*Laboratory for Audio-Visual Communications (LCAV)
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne CH 1015, Switzerland
Email:{guillermo.barrenetxea, baltasar.beferull, martin.vetterli}@epfl.ch
†Department of Electrical Engineering and Computer Sciences
University of California at Berkeley, Berkeley CA 94720, USA

***Abstract*— The analysis and design of routing algorithms for finite buffer networks requires solving the associated queue network problem which is known to be hard. We propose alternative and more accurate approximation models to the usual Jackson's Theorem that give more insight into the effect of routing algorithms on the queue size distributions.**

**Using the proposed approximation models, we analyze and design routing algorithms that minimize overflow losses in grid networks with finite buffers and different communication patterns, namely uniform communication and data gathering. We show that the buffer size required to achieve the maximum possible rate decreases as the network size increases.**

**Motivated by the insight gained in grid networks, we apply the same principles to the design of routing algorithms for random networks with finite buffers that minimize overflow losses. We show that this requires adequately combining shortest path tree routing and traveling salesman routing. Our results show that such specially designed routing algorithms increase the transmitted rate for a given loss probability up to almost three times, on average, with respect to the usual shortest path tree routing.**

## I. Introduction

In many scenarios, packet buffering is expensive in terms of cost, processing and/or space. For instance, common devices used in sensor networks present a limited and generally small amount of memory [1]. This problem is also faced in the context of optical networks, where optical buffering and all-optical processing are still technologically difficult tasks. In this paper, we focus on the analysis and design of routing algorithms that maximize the throughput per node in dense networks with finite buffers, or in other words, algorithms that minimize the overflow losses for a given transmission rate. This analysis requires solving the queueing problem associated to the network. However, no analytically exact solutions are known for even the simplest queueing networks [2] and queueing approximations are required to model the network.

Depending on the purpose of the network (monitoring, data collection, actuation), various traffic patterns can be considered. Particularly, we study two different communication patterns: uniform communication (UC) and central data gathering (CDG). UC corresponds to a distributed control network, where every node needs the information generated by all nodes in the network [3]. CDG represents a monitoring network, where the information generated by all nodes in the network is collected by one node (sink) [4].

We assume that either the sensor network is wired (e.g. CMOS circuits) or, if it is wireless, that there exits a transmission schedule that avoid conflicts which is implemented in the MAC layer. We abstract the wireless case as a graph with point-to-point links and transform the problem into a graph with nearest neighbor connectivity.

We begin with the problem for the case of square grid networks. We propose alternative and notably improved approximation models to the classical Jackson's Theorem, to analyze the distribution on the queue size for the most loaded node where overflow losses will first appear as the transmission rate is increased. These approximation models allow very good analysis in the medium load regime. Using these models, we characterize the optimal routing algorithm that minimize overflow losses, consisting in a traveling salesman (TS) routing. We also show the existing trade-off between overflow losses and delay.

Motivated by the insight gained in grid networks, we apply similar principles to the design of routing algorithms that minimize overflow losses for random networks. In this case, the appropriate routing strategy consist in combining adequately the shortest path tree (SPT) routing and TS. The maximum rate achieved by the proposed algorithm is almost three times the rate achieved with the usual SPT routing for small buffers and dense networks.

The rest of the paper is structured as follows: In Section II, we introduce the network model and assumptions. In Section III, we study the uniform communication pattern in a square grid. In Section IV, we carry out a similar analysis for CDG. In Section V we study the routing problem in random networks. Finally, conclusions are presented in Section VI.

### A. Related Work

Most previous research work on finite buffers is based mainly on Jackson's theorem [2]. Harchol-Balter and Black [5] considered the problem of determining the distribution on the queue sizes induced by the greedy routing algorithm in square grid and torus networks, assuming exponentially distributed service time for the edges. This hypothesis allows the reduction of the problem into a product-form Jackson queue network and its analysis using standard queueing theory techniques. Mitzenmacher [6] also approximated the system using an equivalent Jackson network with constant service time queues. Leighton [7] provided bounds on the the tail of the delay and queue size distributions of the greedy routing algorithm for square grid and torus networks.

## II. Model and Definitions

We consider square grid based sensor networks composed of devices with routing capabilities that generate constant size packets (traffic) as shown in Fig. 1, where edges represent communication links between the nodes. The length of a path is defined as the number of links in that path. A link $l$ 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 a transmission. We denote by $\varphi(d_i)$ the set of links connected to the node $d_i$. For simplicity, we consider the time unit to be the time it takes a packet to traverse one link.

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

0-7803-9201-9/05/$20.00 ©2005 IEEE

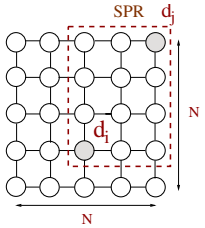



Fig. 1. Network model example: $5 \times 5$ square grid. The shortest path region $\text{SPR}(d_i, d_j)$ between nodes $d_i$ and $d_j$ is delimited by the dashed rectangle.

generate information independently, following a stationary Bernoulli distribution, with a constant average rate of $\mathcal{R}$ packets per time unit. We assume also 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. In this setting, we can consider that each nodes has four queues, each associated with one of the four output links of the node.

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 transmission.

We define network capacity, $C(N)$, as the maximum average number of information packets that can be transmitted reliably per node and per time unit, in a network of size $N \times N$, assuming that the nodes have an infinite buffer.

We denote by $\mathcal{R}^{\Pi}_{\max}(N, Q)$ the maximum average rate that can be transmitted reliably per node and per time unit, in a $N \times N$ grid network, with buffer size $Q$, for a given routing algorithm $\Pi$. Obviously, $\mathcal{R}^{\Pi}_{\max}(N, Q) \leq C(N)$. Strictly speaking, the probability of losing a packet due to buffer overflow under random packet generation is always non-zero for any value of $\mathcal{R} > 0$. Therefore, we consider a transmission to be reliable when the loss probability is smaller than a given threshold. In the subsequent sections, we study routing algorithms that achieve the maximum $\mathcal{R}^{\Pi}_{\max}(N, Q)$ for different communication patterns.

## III. Uniform Communication

The uniform communication pattern models fairly well the scenario of a distributed control network, where the probability of any node communicating to any other node in the network is the same for all pairs of nodes. We start with a brief review of the infinite buffer case and then analyze the effect of finite buffers.

### A. Routing with Infinite Buffers

If finite buffers are not considered, the analysis is only based on stability issues: when the arrival rate is higher than the departure rate, queues become unstable and the expected delay is unbounded. The network capacity $C_u(N)$ is given by [8]:

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

where $c_l$ is equal to 1 for half-duplex and 2 for full-duplex.

Network capacity can be achieved by using an appropriate shortest path routing algorithm $\Pi$ [8]. Note that the bottleneck of the network is clearly located in the central nodes. Intuitively, to maximize the maximum achievable rate per node $\mathcal{R}^{\Pi}_{\max}(N, \infty)$, a routing algorithm has to avoid routing packets through the grid center and promote, as much as possible, the distribution of traffic towards the borders of the grid.

This can be accomplished by the following simple routing strategy: 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. We denote this routing by *row-first* (*column-first*)[7]. Indeed, $\mathcal{R}^{\text{row-first}}_{\max}(N, \infty) = C_u(N)$ [8].

### B. Routing 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 $\mathcal{R}^{\Pi}_{\max}(N, \infty)$. In a square grid, the node located in the center is clearly the most loaded node. We denote it as $d_i$. In this section, we restrict our analysis to $d_i$ and to the routing algorithm that achieves capacity with infinite buffers, that is, row-first.

Computing the network capacity for different buffer sizes $Q$ requires analyzing the associated queue network and computing the distribution on the queue size at $d_i$. However, the analysis of queue networks is complex and no analytically exact solutions are known even for the simplest cases [2]. In this section, we introduce some approximations that simplify the analysis and provide meaningful theoretical results that, as experimentally shown later, are close to the results obtained by simulation.

First, we decompose $d_i$ into four identically distributed and independent FIFO queues associated to its four output links. The input packets to $d_i$ whose final destination is not $d_i$, are sent through one of the four output links depending on their destinations. In view of the symmetry of $d_i$, 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 $d_i$ as the addition of these four iid 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 queue, $q_i$, associated with the output link $l_i$ in $d_i$.

*1) Full-Duplex communication channels:* If $l_i$ is a full-duplex channel, $q_i$ has a dedicated link and it can be modeled as a deterministic service time queue. In this approximation model, we use some results by Neely, Rohrs and Modiano [9], [10], on equivalent models for multi-stage tree networks of deterministic service time queues. We begin by reviewing the main theoretical results in [9], [10], and then show how these results can be applied to our problem.

*Theorem 1:* ([9]) 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 units.

*Theorem 2:* ([10]) 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. 2 shows the equivalence provided by Theorem 1. Fig. 3 shows a tree system and 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 $q_i$. First, we identify $q_i$ as the head node of a tree network composed of

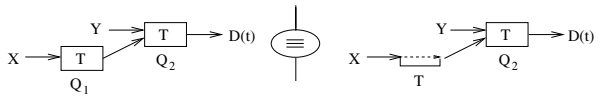



Fig. 2. The total number of packets in a two-queue system remains the same if the first stage queue is replaced by a pure delay of $T$ time units.

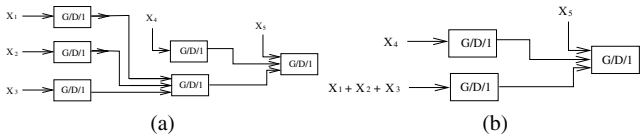



Fig. 3. The number of packets in the head node of the tree network (a) is the same as in the two-stage equivalent model (b).

all the nodes sending traffic through $l_i$ (Fig. 4). Applying Theorem 2, the distribution on the size of $q_i$ can be approximated by the distribution at the head node of the two-stage model (Fig. 4), where we only consider the three neighbors located one hop away from $d_i$ preserving the traffic generated by the entire network that flows through $l_i$.

Note that the tree network associated to $q_i$ (Fig. 4) does not correspond exactly with the tree network of Theorem 2 (Fig. 3). The reason is that, in addition to the exogenous inputs generated at each node, we also have some traffic leaving the network that corresponds to the traffic that reached its destination. Note that the average traffic leaving the network at any node is equal to $\mathcal{R}$. 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 though the same node. Hence, the two-stage model provides an approximated network.

According to Theorem 2, the arrivals to the nodes of the first stage in the two-stage model correspond to the addition of all exogenous inputs routed through $l_i$. 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 packets travel $\mathcal{O}(N)$ hops on average before reaching their destination. Using the row-first algorithm, 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 $d_i$:

$$p_c = \frac{N-1}{N+1}, \quad p_t = \frac{N-1}{2N(N+1)}. \tag{2}$$

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

Apart from the traffic that arrives from its neighbors, $d_i$ generates also new traffic that is injected to the network at a rate $\mathcal{R}$. Considering again the symmetry of $d_i$, the fraction of this traffic that goes through $l_i$ is $\mathcal{R}/4$. The average arrival rate $\lambda_{q_i}$ to $q_i$ can be computed as the addition of both contributions:

$$\lambda_{q_i} = \mathcal{R}/4 + \lambda_1 \left(p_c + 2p_t\right), \tag{3}$$

where $\lambda_1$ is the total arrival rate to the neighbors of $d_i$ (Fig. 4).

For row-first routing, $\lambda_1$ is equal to $\lambda_1 = \mathcal{R}N/4$. We can express $\mathcal{R}$ as a fraction of the network capacity $C(N)$, that is, $\mathcal{R} = \alpha C(N)$, and denote $\alpha$ as relative capacity. Then, using (1), $\lambda_1 = \alpha$.

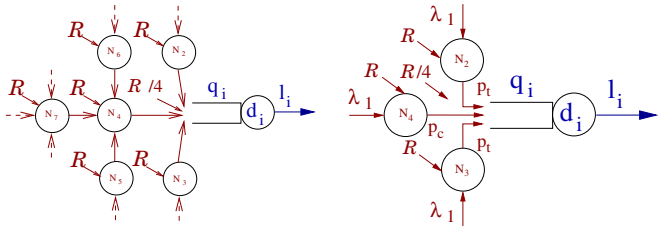



Fig. 4. Tree network approximation: (a) the queue associated to the output link $l_i$ of $d_i$ is the head node and (b) its two-stage model

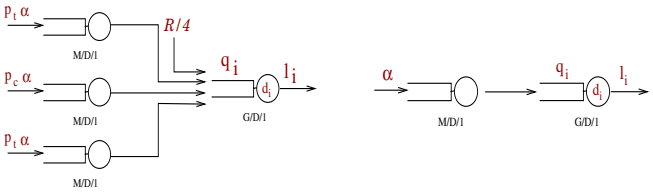



Fig. 5. Approximation models: (a) Two-stage model and the (b) two-queue model.

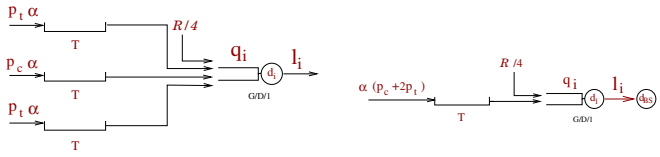
Putting everything together, the resulting approximation model is shown in Fig. 5. Regardless of the number of nodes in the network, we reduce the analysis of the distribution on the size of $q_i$ to a four queue network. This approximation holds for any input traffic distribution as long as it is stationary and independent among the different sources.

*Theorem 3:* The buffer size $Q$ required to achieve a certain relative capacity $\alpha$ decreases with the network size $N$. Furthermore, the required buffer size goes asymptotically to zero.

**Proof:** By Theorem 1, the total number of packets in the approximated model (Fig. 5) is the same as a system where the queues of the first stage have been replaced by pure delays of $T$ time units, and this is equivalent to injecting all the arrivals into a single pure delay (Fig. 6). The total average arrival rate $\lambda_{S_1}$ to the first queues is $\lambda_{S_1} = \alpha(p_c + 2p_t) = \alpha(N-1)/(N+1)$. Therefore, note that for a fixed $\alpha$, the total number of packets in this two queue model is almost constant with $N$.

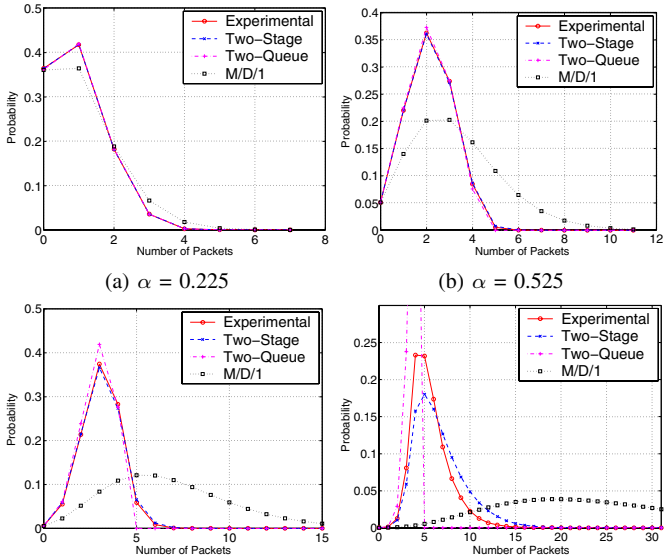
We can decompose the total number of packets $S(t)$ in the approximated model (Fig. 5) as the number of packets in the first stage $S_1(t)$ plus the number of packets in the head node $S_h(t)$. As $N$ increases, $p_c$ goes asymptotically to one and most of the traffic is served by the same first stage queue. Consequently, for a fixed $\alpha$, $S_1(t)$ 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. 5, 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 simplify the model for large networks by assuming that the number of packets turning at $d_i$ is negligible; that is, the packets arrive at $q_i$ only from the neighbor located in the same row or column as $l_i$. Similarly, the exogenous input traffic generated at $d_i$, goes also asymptotically to zero ($\mathcal{O}(1/N)$) as compared with the incoming traffic $\alpha$, and can also be neglected.




(a) (b)

Fig. 6. Equivalent models: (a) we replace the queues of the first stage by pure delays of $T$ time units and (b) we inject all the arrivals to a single pure delay.




(a) $\alpha$ = 0.225 (b) $\alpha$ = 0.525

(c) $\alpha$ = 0.725 (d) $\alpha$ = 0.925

Fig. 7. Distribution on the queue size at $d_i$ for different values of $\alpha$ in a $121 \times 121$ square grid network with full-duplex links.

Consequently, we approximate the queue network by a two-queue model where $q_i$ 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 $\alpha$ (Fig. 5). It follows that the number of packets in $q_i$ is (at most) one with probability $\alpha$ and zero with probability $1 - \alpha$.

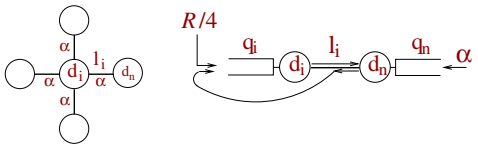
Finally, the distribution $P_i(k)$ on the total queue size $k$ at $d_i$, is given by the addition of four independent and identically distributed queues associate with the four outgoing links from $d_i$:

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

Note that both approximation models proposed (Fig. 5) are asymptotically exact.

Fig. 7 shows the distributions on the size of $q_i$ obtained by simulating the whole queueing network, the two-stage model (Fig. 5), the two-queue model (Fig. 5) and the usual M/D/1 approximation for different values of $\alpha$ in a $121 \times 121$ square grid network. 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 [6].

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, we neglected the traffic leaving the network at each (destination) node, which increases as $\mathcal{R}$ increases. On the other hand, the M/D/1 model based on Jackson's theorem, only approximates the distribution under low load conditions. For $\alpha = 0.525$, we already observe that this approximation is far form the distribution obtained experimentally. Under medium and high load conditions, the independence assumption does not hold and the approximation quality degrades rapidly.




(a) (b)

Fig. 8. Half-duplex links: (a) central node and (b) approximated model for the queue associated to $l_i$.

Fig. 10 shows the distribution on the queue size at $d_i$, 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 the size of the network $N$ increases and the packet distribution converges to the two tandem queues model (Fig. 5).

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

To capture the dependence between arrivals and departures, we propose the following approximation model. Every time $d_i$ wants to send a packet through $l_i$, it has to compete for $l_i$ with one of its neighbors, $d_n$ (Fig. 8). If $d_i$ takes $l_i$ first, it can transmit a packet and the size of $q_i$ is reduced by one. However, if $d_n$ takes the link first and sends a packet, not only is $d_i$ unable to transmit, but also the size of $q_i$ is increased by one if the final destination of the packet is not $d_i$.

Note that, in practice, packets sent by $d_n$ never go through $l_i$ (packets do not go backwards) although they stay in $d_i$. However, notice that by putting these packets into $l_i$, we simulate packets arriving from the other neighbors of $d_i$ and prevent packet transmissions. This approximation is represented in Fig. 8.

We denote by $\rho_i$ the utilization factor of $q_i$. That is,

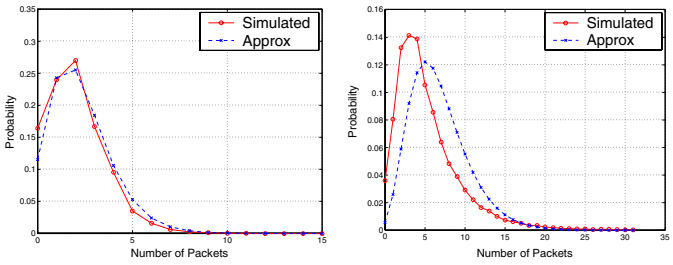
$$\rho_i = \frac{\lambda_{q_i}}{\mu_{q_i}} = \frac{\alpha}{\mu_{q_i}},$$

where $\lambda_{q_i}$ is the arrival rate to $q_i$, and $\mu_{q_i}$ is the service rate. Note that $\lambda_{q_i}$ is identical in both half-duplex and full-duplex models.

Similarly, we denote by $\rho_n$ the utilization factor of the queue $q_n$ in $d_n$ associated to $l_i$. We assume that $d_i$ and $d_n$ have the same probability to capture the link for a transmission. Therefore, if $q_i$ has a packet waiting to be transmitted, the probability $p_s$ of sending it in this time unit is simply equal to the probability of $d_i$ being the first to capture the link plus the probability of $d_n$ having nothing to transmit through $l_i$:

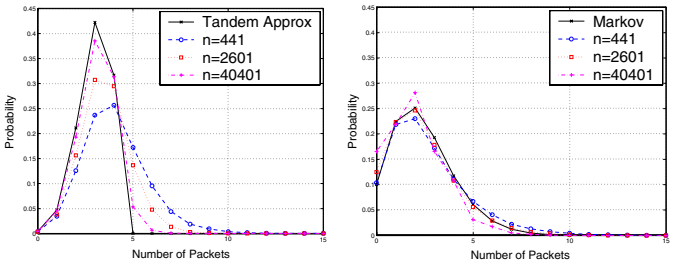
$$p_s = \frac{1}{2} + \frac{1}{2}(1 - \rho_n) = 1 - \rho_n/2; \quad (5)$$

We assume that if $d_i$ does not capture $l_i$ for a transmission in this time unit, it tries to capture it again in the next time unit. Therefore,




(a) $\alpha$ = 0.725 (b) $\alpha$ = 0.925

Fig. 9. Distribution on the queue size at $d_i$ obtained by simulation and with the Markov chain approximation for different values of $\alpha$ in a 121 × 121 grid network with half-duplex links.




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

Fig. 10. Distribution on the queue size at $d_i$ for a constant relative capacity $\alpha = 0.75$ and different network sizes with (a) full-duplex and (b) half-duplex links.

we model the service time of $q_i$ as a geometric distribution with parameter $p_s$.

As in the full-duplex case, we approximate arrivals to $d_n$ as a Poisson distribution with parameter $\alpha$. In addition to arrivals from $d_n$, new packets are also generated at $d_i$ with rate $\mathcal{R}$. Considering again the symmetry of $d_i$, the fraction of this traffic that goes through $l_i$ is $\mathcal{R}/4$. Note that arrivals and service time distributions are both memoryless, and therefore, if we denote by $X_i(t)$ the number of packets in the queue $q_i$ at time $t$, $\{X_i(t) \mid t > 0\}$ can be modeled using a Markov chain. As the network size increases, the difference between both utilization factors, $\rho_i$ and $\rho_n$, becomes negligible, and we can assume that $\rho_i = \rho_n = \rho$. Moreover, the new traffic generated at $d_i$ becomes also asymptotically negligible ($\mathcal{O}(1/N)$) compared to the traffic that arrives from $d_n$. Applying both simplifications, the transition probability matrix $P_i(j,k)$, associated with $\{X_i(t)|t > 0\}$, can be approximated by:

$$P_i(0,k) = \begin{array}{ll} 1-\rho, & k=0, \\ \rho, & k=1, \end{array} \qquad P_i(j,k) = \begin{array}{ll} 1-\frac{\rho}{2}, & k=j-1, \\ \frac{\rho}{2}, & k=j+1. \end{array}$$

Fig. 9 shows the distributions on the queue size at $d_i$ for different values of $\alpha$ in a 121 × 121 square grid network with half-duplex links. This model closely approximates the experimental distribution for low to moderate rate per node ($\alpha < 0.8$), while the approximation quality degrades when the traffic is higher.

Fig. 10 shows the distribution on the queue size at $d_i$ for a constant relative capacity $\alpha = 0.75$ as a function of the network size.

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, $l_i$ is shared between $d_i$ and $d_n$ and, even if the input rate $\lambda_{l_i}$ is less than the link capacity, there is a non-zero probability that $d_i$ competes for the link with $d_n$, in which case one of them has to store the packet for a further transmission.

## IV. Central Data Gathering

In central data gathering, every node transmits information to a particular and previously designated single node $d_{BS}$, denoted *base station*, that can be located anywhere in the network. We start with a brief review of the infinite buffer case and then analyze the effect of finite buffers.

### A. Routing with Infinite Buffers

Under the infinite buffer hypothesis, the network capacity $C_{cdg}(N)$ can be easily obtained based on stability issues [8]:

$$C_{cdg}(N) \leq \frac{|\varphi(d_{BS})|}{N^2-1}, \tag{6}$$

where $|\varphi(d_{BS})|$ denotes the cardinality of $\varphi(d_{BS})$.

The necessary and sufficient condition for a routing algorithm $\Pi$ to achieve capacity is that $\Pi$ distributes the total arrival traffic to $d_{BS}$ uniformly among the links in $\varphi(d_{BS})$. For the sake of simplicity, we restrict our analysis to a particular location of $d_{BS}$: the grid center. Nevertheless, a similar analysis can be carried out for any location. In this case, a simple routing algorithm that achieves capacity with infinite buffers is the *random greedy algorithm* [7]: for each packet, nodes use a row-first or a column-first routing algorithm with equal probability.

### B. Routing with Finite Buffers

Although there are many routing algorithms that achieve capacity under the infinite buffer hypothesis, we show in this section that their performance is quite different when the buffers are constrained to be finite. To analyze the network capacity for a given routing algorithm under finite buffers, we proceed as in the UC case. First, we identify the most loaded node $d_i$ and associate the network to a tree. Then, we reduce this tree to its two-stage equivalent model and obtain the packet distribution in $d_i$ by analyzing the packet distribution in the head node of the two-stage model. We perform this analysis for any routing algorithm $\Pi$ that achieves capacity under infinite buffers.

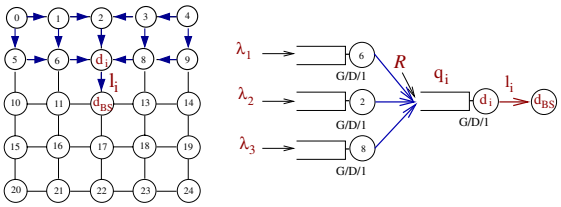
The bottleneck of the network is clearly located in the neighbors of $d_{BS}$. Moreover, if $\Pi$ achieves capacity for infinite buffers, the total arrival traffic to $d_{BS}$ is uniformly distributed among the links in $\varphi(d_{BS})$. Due to the independence of packet generation, the distributions on the queue size in these four nodes are iid. Consequently, we reduce the problem to computing the queue distribution for only one of these neighbors, $d_i$. We denote by $l_i$ the link between $d_i$ and $d_{BS}$, and by $q_i$, the queue in $d_i$ associated to $l_i$ (Fig. 11).

We consider now only those nodes that generate traffic through $l_i$. These nodes form a tree with $q_i$ as head, with exogenous inputs at each node and with no traffic leaving the network. Applying Theorem 2, the packet distribution in $q_i$ is the same as in its two-stage model (Fig. 11). A key point is that in this case, the two-stage model is not an approximation but an exact model for any rate.

The arrivals to the three nodes of the first stage are the addition of all the traffic generated by the network that goes through $l_i$. If $\Pi$ achieves capacity for infinite buffers, the total average traffic that flows through $l_i$ is equal to:

$$\lambda_{l_i} = \frac{\mathcal{R}(N^2-1)}{4}. \tag{7}$$

We denote by $\lambda_1$, $\lambda_2$ and $\lambda_3$ the average arrival rate to the three first stage nodes of the two-stage model (Fig. 11). These three nodes have




(a) (b)

Fig. 11. Tree network in CDG: (a) tree where $l_i$ is the head node and (b) its two-stage equivalent model.

to route all the traffic that goes through $l_i$ except the traffic generated by $d_i$ itself. That is,

$$\lambda_1 + \lambda_2 + \lambda_3 = \frac{\mathcal{R}(N^2-1)}{4} - \mathcal{R}. \quad (8)$$

We obtain the distribution on the size of $q_i$ by analyzing the distribution at the head node of the two-stage model. Particularly, we are interested in finding the routing algorithm $\Pi$ that, for a given $Q$, achieves the maximum rate per node $\mathcal{R}^{\Pi}_{max}(N,Q)$. This is equivalent to minimizing the number of packets in $q_i$ for a given $\mathcal{R}$.

*Lemma 1:* 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 $\lambda_i$ 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:

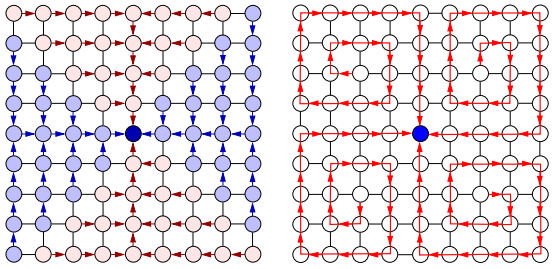
$$\lambda_i = \begin{cases} \lambda_t, & \text{for } i\text{=1,2 or 3,} \\ 0, & \text{otherwise.} \end{cases}$$

**Proof:** By Theorem 1, the total number of packets in the two-stage model (Fig. 11) 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 into a single pure delay(Fig. 6). Consequently, the total number of packets in the system is equivalent for any combination of $\lambda_i$ 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. ■

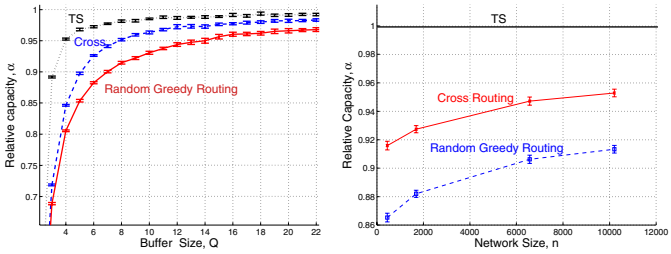
Consequently, the routing algorithm that achieves the maximum $\mathcal{R}^{\Pi}_{max}(N,Q)$ is such that the input traffic to $d_i$ arrives only from one of its neighbors. However, the congestion problem is now translated to this neighbor of $d_i$. Furthermore, as the network size increases, the difference between the traffic that flows through $d_i$ and its neighbor goes asymptotically to zero. To solve this, we apply Lemma 1 recursively, that is, the optimal routing algorithm is such that all nodes receive as much of their traffic as possible from only one neighbor.

The shortest path routing algorithm that implements this principle is shown in Fig. 12 and consists in the following. In the $N\times N$ square grid, 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 square grid. 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 forwarding 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 1.



(a) Cross routing (b) TS routing

Fig. 12. CDG routing: (a) Cross routing, the optimal shortest path routing and (b) a TS routing algorithm, the optimal non-shortest path routing.



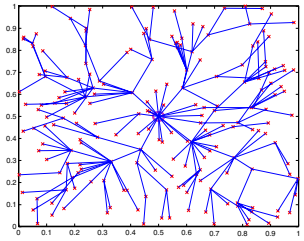

(a) (b)

Fig. 13. Routing for CDG: (a) maximum relative capacity achieved by different routing algorithms in a $21\times 21$ square grid for different buffer sizes $Q$, (b) performance of cross routing and greedy routing relative to TS routing for a fixed buffer size $Q=5$ for different network sizes $n$, both plots with 95% CI.

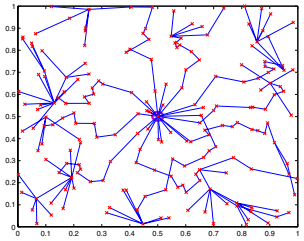
According to Lemma 1, the optimal routing consists in making nodes receive all traffic exclusively from one neighbor. This condition can only be fully satisfied by non-shortest path routing algorithms. Applying Lemma 1 recursively, the set of optimal routing algorithms is such that it divides the network 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 these optimal routing algorithms as traveling salesman (TS) routing. Fig. 12 shows an example of a TS routing algorithm. Clearly, TS routing algorithms generate the optimal arrival distribution in all nodes.

Although TS routing achieves the maximum $\mathcal{R}^{\Pi}_{max}(N,Q)$, the delay incurred by the packets may be unacceptable. Notice that the average path length $\overline{L}_{TS}$ for any TS routing algorithm is $\mathcal{O}(N^2)$, while for any shortest path routing, $\overline{L}_{s-p}$ is $\mathcal{O}(N)$. TS 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. 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.

We compare the performance of random greedy routing, cross routing and TS routing. Fig. 13 shows the maximum relative capacity $\mathcal{R}^{\Pi}_{max}(N,Q)/C_{cdg}(N)$ achieved by different routing algorithms in a $21\times 21$ square grid network as a function of the buffer size $Q$, with the 95% confidence intervals (CI). Notice that although all routing algorithms asymptotically achieve capacity as the buffer size

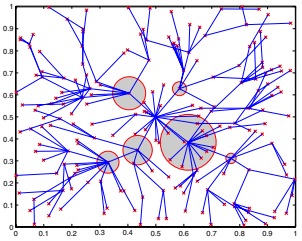


(a) SPT

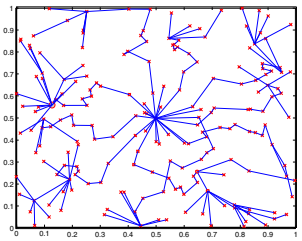


(b) TS-SPT

Fig. 14. Central data gathering routing in a random network: routes generated by (a) shortest path tree and (b) TS-SPT routing algorithms.



(a) SPT routing losses



(b) TS-SPT routing losses

Fig. 15. Overflow losses per node with (a) SPT routing and (b) TS-SPT routing. The circles around nodes have a radius proportional to the overflow losses in that node.

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 TS routing, while cross routing performs best among shortest path routing algorithms.

Fig. 13 shows also the maximum rate achieved by different routing algorithms relative to the maximum rate achieved by TS routing for a fixed $Q = 5$, as a function of the network size $N$, with the 95% CI. Since all routing algorithms analyzed are asymptotically optimal with the network size, the performance gap between TS routing and these algorithms decreases as the network size increases for a fixed value of $Q$.

## V. Data Gathering in Random Networks

Motivated by the insight gained in the grid based networks, we extend now our results to random networks. For the square grid, we showed that the routing algorithm that minimizes overflow losses for central data gathering, consists in distributing the load uniformly among the four arrival links to the base-station and uses a TS routing algorithm within the set of nodes associated to each of these links. The reason is that overflow losses are higher in nodes receiving traffic from several neighbors, being more critical for those nodes close to the base-station, and consequently, in a higher load regime.

We now examine a different scenario. We randomly place $n$ nodes on a unit square surface following a uniform distribution. We consider a simple boolean model with a circular connectivity range $R_C$ for each node, where $R_C$ is constant for all nodes. That is, if the distance between two nodes is less than the communication radio, we assume that there is a link between both nodes. We assume also that there exits a transmission schedule that avoid conflicts, so that we abstract the random network as a random graph with point-to-point links. We consider a CDG scenario where, for simplicity, we locate the base-station $d_i$ in the center of the square surface.

Even if random networks are very different in nature from grid networks, as we show next, the same principles can be applied to the design of routing algorithms. The critical nodes are also those located close to $d_i$, and to minimize overflow losses, they have to receive traffic from only one of their neighbors. In other words, these nodes have to route packets using a TS routing algorithm. However, to define a TS routing in the network such that all nodes remain connected, is hard to solve in a distributed way.

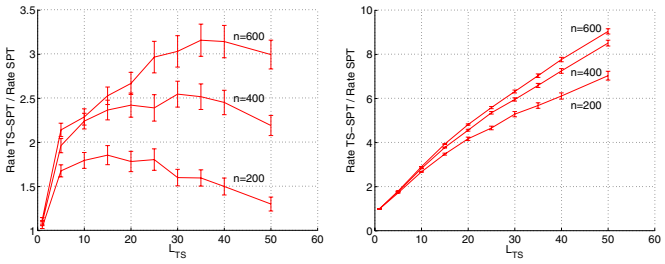
Note that nodes located far from the base-station, and consequently, carrying less traffic, are less critical for overflow losses. Based on these principles, we propose the following routing algorithm. First, we establish $|\varphi(d_i)|$ disjoint TS routes of certain length $L_{TS}$ departing from each neighbor of $d_i$. To construct the TS routes, we use a simple heuristic algorithm: we select as next node of the route the closest node, if any, that does not belong to any TS route. Second, the nodes that do not belong to any TS route, direct their traffic preferentially to the end points of the TS routes using a shortest path routing. To do so, we use a modified Bellman-Ford algorithm: the nodes belonging to any TS route set their effective distance to $d_i$ according to their position in the route, such that, its effective distance is inversely proportional to the number of hops towards $d_i$. Consequently, there is a big penalty to use those nodes close to $d_i$ belonging to any TS route. Using this penalty gradient, we ensure that all nodes remain connected. We denote this routing algorithms as TS-SPT routing. Note that the TS-SPT routing algorithm is executed in a totally distributed way. Fig. 14 shows the routes used to transmit the information to $d_i$ using shortest path tree (SPT) and TS-SPT routing algorithms with $L_{TS} = 5$ in a $n = 200$ random network. Note that both trees select a subgraph of the original connectivity graph.

Fig. 15 shows the overflow losses per node using a SPT and TS-SPT routing algorithms for a fixed transmission rate per node, in the random network shown in Fig. 14 when we limit the buffer size in the nodes to $Q = 2$. The circles around nodes have a radius proportional to the overflow losses in each node. As expected, SPT routing induces losses in those nodes close to $d_i$ that receive traffic from multiple neighbors. As can be seen in Fig. 15, these hot-spots are suppressed when TS-SPT routing is used instead.

Note that to reduce overflow losses, it is convenient to choose a large value for $L_{TS}$, so that many nodes receive traffic from only one node. However, if $L_{TS}$ is too large, the traffic would be distributed unevenly among the neighbors of $d_i$ and overflow loses would also consequently increase. This suggests that there exits an optimal value for $L_{TS}$ for which overflow losses are minimized. Note that the number of hops toward $d_i$ increases linearly with $L_{TS}$. Therefore, as in the square grid, there exist a trade-off between overflow losses and the average number of transmissions.

To analyze the performance of TS-SPT, we carry out the following experiment. We distribute $n = [200, 400, 600]$ nodes with a fixed communication range $R_c = 0.17$, so that we can almost guarantee that all nodes are connected. Note that, as we increase $n$, we increase the density of the network. For several network realizations, we analyze the maximum rate per node achieved for a maximum overflow loss probability given, using both, SPT and TS-SPT routing algorithms, when the buffer size per node is limited to $Q = 2$.

Fig. 16 depicts the maximum rate per node archived for a given loss probability and the average number of hops required to transmit a packet to $d_i$ by TS-SPT relative to SPT. TS-SPT always achieves a higher rate per node that, on average, can be up to almost three




(a) (b)

Fig. 16. TS-SPT routing performance with respect to STP routing: (a) maximum rate achieved for a given overflow loss probability and (b) average number of hops towards the base-station for 150 random networks with the 95% CI.

times the rate achieved with SPT routing. The gain is clearly more noticeable for dense networks. The reason is that, in dense networks, nodes located close to the base-station are in the communication range of an increasing number of nodes, and consequently, they receive traffic from multiple nodes. This is the case where the use of TS routes close to $d_i$ gives a higher gain. On the other hand, note also that, as expected, the number of hops increases almost linearly with $L_{TS}$. This illustrates the trade-off between overflow losses and the average number of transmissions.

## VI. Conclusions

In this paper, we first analyze optimal routing algorithms that minimize overflow losses in grid networks. We present TS routing as the extreme case of the existing trade-off between overflow losses and transmissions: it achieves the maximum throughput per node while the number of required transmissions increases drastically. For random networks, we present also several results which indicate that the intuition gained from grid networks is valuable for the design of routing algorithms that also trade-off overflow losses and transmissions. Our current research focuses on extending these results to include important practical issues, such as energy constrains.

Usually, the information that nodes send to the base-station is highly correlated. This correlation can be exploited to reduce the rate that nodes inject into the network. Particularly, we are interested in studying the interaction of the routing problem for finite buffers and the correlation in the data.

## Acknowledgment

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.

## References

[1] Crossbow Products: Wireless sensor networks, "http://www.xbow.com"
[2] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall International Editions, 1992.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures*, Morgan-Kaufman, 1991.
[8] G. Barrenetxea, B. Beferull-Lozano, and M. Vetterli, "Lattice networks: Capacity limits, optimal routing and queueing behavior," *Submitted to IEEE/ACM Transactions on Networking*.
[9] 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.
[10] 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.