[10] S. A. Borbash, "Design Considerations in Wireless Sensor Networks," Ph.D. dissertation, Univ. Maryland, College Park, MD, 2004.
[11] R. C. Merkle and M. E. Hellman, "Hiding information and signatures in trap door knapsacks," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 5, pp. 525–530, Sep. 1978.
[12] H. Karloff, *Linear Programming*. New York: Springer Verlag, 1991.
[13] E. Scheinerman and D. Ullman, *Fractional Graph Theory*. New York: Wiley, 1997.
[14] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.
[15] S. A. Borbash and A. Ephremides, "Wireless link scheduling with power control," in *Proc. WiOpt*, 2004.
[16] V. Y. Pan and Z. Q. Chen, "The complexity of the matrix eigenproblem," in *Proc. ACM Symp. Theory Comput.*, 1999.

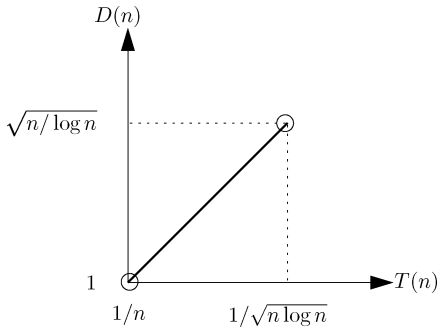

Fig. 1. Throughput–delay scaling tradeoff in the static random network model. The scales of the axes are in terms of orders in n.

Optimal Throughput–Delay Scaling in Wireless Networks—Part II: Constant-Size Packets

Abbas El Gamal, *Fellow, IEEE*,
James Mammen, *Student Member, IEEE*,
Balaji Prabhakar, *Senior Member, IEEE*, and Devavrat Shah

***Abstract*—In Part I of this paper, the optimal throughput–delay tradeoff for static wireless networks was shown to be $D(n) = \Theta(nT(n))$, where $D(n)$ and $T(n)$ are the average packet delay and throughput in a network of n nodes, respectively. While this tradeoff captures the essential network dynamics, packets need to scale down with the network size. In this "fluid model," no buffers are required. Due to this packet scaling, $D(n)$ does not correspond to the average delay per bit. This leads to the question whether the tradeoff remains the same when the packet size is kept constant, which necessitates packet scheduling in the network.**

In this correspondence, this question is answered in the affirmative by showing that the optimal throughput–delay tradeoff is still $D(n) = \Theta(nT(n))$, where now $D(n)$ is the average delay per bit. Packets of constant size necessitate the use of buffers in the network, which in turn requires scheduling packet transmissions in a discrete-time queuing network and analyzing the corresponding delay. Our method consists of deriving packet schedules in the discrete-time network by devising a corresponding continuous-time network and then analyzing the delay induced in the actual discrete network using results from queuing theory for continuous-time networks.

***Index Terms*—Product form equilibrium, queuing theory, scaling laws, scheduling, throughput scaling, throughput–delay tradeoff, wireless networks.**

I. Introduction

In their seminal work [4], Gupta and Kumar introduced a random network model for studying throughput scaling in a static wireless network, i.e., when the nodes do not move. They showed that the throughput per source–destination (S–D) pair scales as $\Theta\left(1/\sqrt{n\log n}\right)$. They implicitly used a fluid model in which the packet size is allowed to be arbitrarily small. Later work by Kulkarni and Viswanath [6] consolidated the result with an explicit constant packet size model.

In previous work [1], we studied the throughput–delay tradeoff in wireless networks. A more complete treatment is provided in Part I [2] of this two-part work. The optimal throughput–delay tradeoff is established to be $D(n) = \Theta(nT(n))$ (see Fig. 1). In this work, packet size needs to scale down with the number of nodes n in the network. This leads to a fluid model for transmitting packets and allows us to obtain the essential tradeoff by skirting the issue of buffering and the resultant queuing delay at the nodes. The delay that is considered in [2] is the average packet delay and since the packet size is allowed to scale down with n, it does not correspond to the average delay per bit. This correspondence investigates the throughput–delay tradeoff when the packet size remains constant, i.e., does not scale down with n. This is an important question, since in real networks, packet sizes do not change when more nodes are added to the network. Note that with the additional constraint that the packet size remains constant, the throughput–delay tradeoff can be no better than that in the fluid model. However, *a priori*, it is not clear whether the same throughput–delay tradeoff as in the fluid case can be achieved, since now, routing packets through the network also involves the additional task of scheduling in the network. In [8], it was shown that in a mobile network model with independent and identically distributed (i.i.d.) mobility (each node is distributed uniformly at random in each time slot independent of others and the past), a two-hop scheme like the one in [3] achieves the optimal tradeoff using packets of constant size. However, this method does not extend to static networks or mobile networks with non-i.i.d. mobility. In this correspondence, we extend our previous work to the case of static wireless networks with buffers and constant-size packets and show that the optimal tradeoff is still $D(n) = \Theta(nT(n))$ (as shown in Fig. 1), where now $D(n)$ is the average delay per bit.

The main contribution of this correspondence is a scheduling policy for which it is shown that the throughput–delay tradeoff is the same as that for the fluid model. Analyzing the delay of any scheduling policy for a wireless network corresponds to analyzing the delay of an induced discrete-time queuing network. It is natural to attempt to use a first-in-first-out (FIFO) queue management in the wireless network, however, not much is known about delay with FIFO in discrete-time queuing networks. Thus, the study of achievable throughput–delay tradeoff with packets of constant size requires a scheduling policy with good performance that is amenable to analysis. We provide a solution by coupling

Manuscript received February 9, 2006; revised July 21, 2006. This work was supported by the Stanford Networking Research Center. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, Adelaide, Australia, September 2005.

A. El Gamal and J. Mammen are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: abbas@stanford.edu; jmammen@stanford.edu).

B. Prabhakar is with the Departments of Electrical Engineering and Computer Science, Stanford University, Stanford, CA 94305 USA (e-mail: balaji@stanford.edu).

D. Shah is with the Departments of Electrical Engineering and Computer Science and ESD, Massachusetts Institute of Technology, Cambridge MA 02139-4307, USA (e-mail: devavrat@mit.edu).

Communicated by P. Viswanath, Associate Editor for Communications.

Digital Object Identifier 10.1109/TIT.2006.883548

the evolution of a discrete-time queuing network with that of a continuous-time queuing network. This leads to both a packet scheduling policy (see item 6) of Policy Σ_n in Section II) for the wireless network and a method for analyzing the delay. The following is an outline of our solution. Packets in a wireless network have fixed routes depending on the S–D pair to which they belong. The entire wireless network then corresponds to a discrete-time, open queuing network with general customer routes, in the terminology of queuing theory (e.g., see [5], [9]). In the case of continuous-time queuing networks, when some more conditions are satisfied (such as independent Poisson arrivals to each customer route and a symmetric queue at each server) these are known as Kelly or Baskett–Chandy–Muntz–Palacios (BCMP) networks. For such networks, the equilibrium distribution is known to have a product form. We consider a continuous-time queuing network with general customer routes with the same topology as the discrete-time network we wish to study. Further, this network is assumed to have Poisson arrivals, constant service time and preemptive last-in first-out (LIFO) at each server so that it is a Kelly (BCMP) network. Then, based on packet arrival times in this continuous-time queuing network at each server, we derive a scheduling policy for the discrete-time wireless network. Finally, using product form equilibrium results for continuous-time networks, we determine the exact order of queuing delay in the discrete-time wireless network.

A. Model and Definitions

For the sake of completeness, we repeat the models and definitions already presented in [2]. Refer to [2] for a more complete explanation of the model.

Definition 1 (Static Random Network Model): The static random network consists of a unit torus in which n nodes are distributed uniformly at random. These n nodes are split into $n/2$ distinct S–D pairs at random. Time is slotted for packetized transmission. For simplicity, we assume that the time slots are of unit length.

Definition 2 (Model for Successful Transmission): Under the *Relaxed Protocol* model, a transmission from node i to node j in a time slot is successful if for any other node k that is transmitting simultaneously

$$d(k,j) \geq (1+\Delta)d(i,j), \qquad \text{for } \Delta > 0$$

where $d(i,j)$ is the distance between nodes i and j. During a successful transmission, nodes send data at a constant rate of W bits per second.

With time slots of unit length, this means that the size of packets transmitted in each slot is W bits.

Definition 3 (Scheme): A scheme Π, for a random network, is a sequence of communication policies, (Π_n), where policy Π_n determines how communication occurs in a network of n nodes.

Definition 4 (Throughput of a Scheme): Let $B_{\Pi_n}(i,t)$ be the number of bits of S–D pair $i, 1 \leq i \leq n/2$, transferred in t time slots under policy Π_n. Note that this could be a random quantity for a given realization of the network. Scheme Π is said to have throughput $T_\Pi(n)$ if there exists a sequence of events $A_\Pi(n)$ such that

$$A_\Pi(n) = \left\{ \min_{1\leq i\leq n/2} \liminf_{t\to\infty} \frac{1}{t} B_{\Pi_n}(i,t) \geq T_\Pi(n) \right\}$$

and $P(A_\Pi(n)) \to 1$ as $n \to \infty$.

We use the term *whp* (with high probability) to denote this. That is, we say that an event A_n occurs with high probability (*whp*) if $P(A_n) \to 1$ as $n \to \infty$.

Definition 5 (Delay of a Scheme): The delay of a bit is the time it takes for the bit to reach its destination after it leaves the source. Let $D^i_{\Pi_n}(j)$ denote the delay of bit j of the S–D pair i under policy Π_n, then the sample mean of delay (over packets that reach their destinations) for S–D pair i is

$$\bar{D}^i_{\Pi_n} = \limsup_{k\to\infty} \frac{1}{k} \sum_{j=1}^{k} D^i_{\Pi_n}(j).$$

The average delay over all S–D pairs for a particular realization of the random network is then

$$\bar{D}_{\Pi_n} = \frac{2}{n} \sum_{i=1}^{n/2} \bar{D}^i_{\Pi_n}.$$

The delay for a scheme Π is the expectation of the average delay over all S–D pairs, i.e.,

$$D_\Pi(n) = E[\bar{D}_{\Pi_n}] = \frac{2}{n} \sum_{i=1}^{n/2} E[\bar{D}^i_{\Pi_n}].$$

Definition 6 (Throughput–Delay (T–D) Optimality): A pair $(T(n), D(n))$ is said to be throughput–delay (T–D) optimal if there exists a scheme Π with $T_\Pi(n) = \Theta(T(n))$ and $D_\Pi(n) = \Theta(D(n))$ and $\forall$ scheme Π' with $T_{\Pi'}(n) = \Omega(T(n))$, $D(\Pi')(n) = \Omega(D(n))$.

Definition 7 (Optimal Througput–Delay (T–D) Tradeoff): The optimal T–D tradeoff consists of all the T-D optimal pairs.

Note that in the definition of delay we used bit delay whereas in the scheme we present later, we refer to packet delay. Since the packet size is constant, however, these two are of the same order.

In this correspondence, we use c_i to denote constants that do not depend on n.

Our main result is as follows.

Theorem 1: The optimal T–D tradeoff in the static random network model is given by

$$T(n) = \Theta\left(D(n)/n\right)$$

for $T(n) = O\left(1/\sqrt{n \log n}\right)$.

The preceding result says that under a delay scaling constraint of $D(n)$ the optimal throughput scaling is $\Theta(D(n)/n)$. And this holds for $T(n) = O\left(1/\sqrt{n \log n}\right)$, that is, the entire range of achievable throughputs in the static random network model.

We would like to note that Part I of this work also deals with mobile networks with a random-walk mobility model, in addition to static networks. This correspondence, however, only deals with static networks. This is due to the fact that when nodes are mobile, the wireless network does not correspond to a standard queuing network. Section III contains further discussion on mobile networks with constant-size packets.

The rest of this correspondence is organized as follows. In Section II, we introduce Scheme Π and show that it achieves the T–D tradeoff stated in Theorem 1. Finally, we present a converse that shows that no scheme can provide a better T–D tradeoff than Scheme Π, thus establishing Theorem 1.

II. T–D Tradeoff in Static Networks

Our tradeoff scheme is a multihop, time-division-multiplexed (TDM), cellular scheme with square cells of area $a(n)$ so that the unit torus consists of $1/a(n)$ cells as shown in Fig. 2. In the following analysis, we ignore the edge effects due to $1/a(n)$ not being a perfect

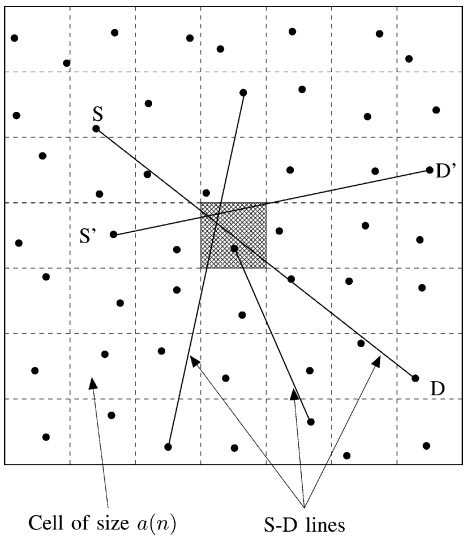

Fig. 2. The unit torus is divided into cells of size $a(n)$ for Scheme Π. The S–D lines passing through the shaded cell in the center are shown.

square. Before presenting the tradeoff scheme, we present three lemmas about the geometry of the n nodes on the torus divided into square cells of area $a(n)$. See [2] for the proofs.

Lemma 1: If $a(n) \geq 2\log n/n$, then each cell has at least one node *whp*.

We say that cell B *interferes* with another cell A if a transmission by a node in cell B can affect the success of a simultaneous transmission by a node in cell A.

Lemma 2: Under the Relaxed Protocol model, the number of cells that interfere with any given cell is bounded above by a constant c_1, independent of n.

We say that a cell is *active* in a time slot if any of its nodes transmits in that time slot. A consequence of Lemma 2 is that there exists an interference-free schedule where each cell becomes active regularly, once in $1 + c_1$ time slots and no cell interferes with any other simultaneously transmitting cell.

Let the straight line connecting a source S to its destination D be called an S–D line.

Lemma 3: The number of S–D lines passing through each cell is $O(n\sqrt{a(n)})$, *whp*.

The preceding lemma shows that the number of S–D lines passing through each cell is $\leq c_2 n\sqrt{a(n)}$ *whp*, for an appropriate choice of the constant c_2.

Now we are ready to describe Scheme Π, which is parameterized by the cell area $a(n)$ where $a(n) = \Omega(\log n/n)$ and $a(n) \leq 1$. Recall that by definition, Scheme Π is a sequence of communication policies (Π_n). For any particular realization of the random network with n nodes, policy Π_n differs based on the following two conditions.

Condition A: No cell is empty.
Condition B: The number of S–D lines through each cell is at most $c_2 n\sqrt{a(n)}$.

If both the above conditions are satisfied then Π_n is the policy Σ_n, described below. Otherwise, Π_n is a time-division policy where each of the $n/2$ sources transmits directly to its destination in a round-robin fashion.

Policy Σ_n:

1) Divide the unit torus using a square grid into square cells, each of area $a(n)$ (see Fig. 2).
2) Each node generates packets according to a Poisson process of rate $T(n) = \Theta\left(1/n\sqrt{a(n)}\right)$. The random network is a discrete-time system whereas the packet generation is a continuous-time process. So if a packet is generated at time t, it is available for transmission from time slot $\lceil t \rceil$ onwards.
3) Each cell becomes active at a regular interval of $1 + c_1$ time slots (see Lemma 2). Several cells which are sufficiently far apart become active simultaneously. Thus, the scheme uses TDM between nearby cells.
4) A source S sends packets to its destination D by relaying or hopping along the adjacent cells lying on its S–D line as shown in Fig. 2. Thus, in this scheme, direct transmission of packets is only between nodes in adjacent cells.
5) One of the nodes in a cell acts as a relay by maintaining a buffer for the packets of all the S–D lines passing through that cell. In each time slot, only one packet can be transmitted. However, a relay node may receive up to four packets from its adjacent cells before it gets a chance to relay them. Moreover, multiple packets may be generated within the cell which will be available for transmission in the next time slot. Hence, a virtual queue is formed in each cell which consists of packets generated within the cell as well as the packets to be relayed through the cell.
6) When the cell becomes active, one packet from this virtual queue (if not empty) is transmitted to an adjacent cell according to a LIFO type of queue service policy. However, the arrival times considered by this policy are not the actual arrival times of the packets, but the arrival times that would occur in a continuous-time network with the same arrivals and a preemptive LIFO (PL) queue management at each server. This is elaborated later in this section during the analysis of delay.

Note that each cell has a single relay node and that it maintains a buffer for all packets of all S–D pairs passing through that cell except for the packets generated by source nodes within the cell. However, the virtual queue in the cell includes these latter type of packets, although, the source nodes do not transmit these packets to the relay node in the cell. We assume that there is coordination within the cell to allow this. As a result, the delay analysis only needs to consider this virtual queue.

The point of tradeoff at which Scheme Π operates is determined by the parameter $a(n)$ and the dependence is made precise in the following theorem.

Theorem 2: For $a(n) = \Omega(\log n/n)$

$$T(n) = \Theta\left(1/n\sqrt{a(n)}\right) \quad \text{and} \quad D(n) = \Theta\left(1/\sqrt{a(n)}\right)$$

i.e., the T–D tradeoff achieved by Scheme Π is

$$T(n) = \Theta\left(D(n)/n\right).$$

Throughput Analysis: If the time-division policy with direct transmission is used, then the throughput is $2W/n$ and delay of 1. But since it happens with a vanishingly low probability, as shown by Lemmas 1 and 3, the throughput and delay for Scheme Π are determined by that of policy Σ_n.

When policy Σ_n is used, since Condition A is satisfied, each cell has at least one node. This guarantees that each source can send data to its destination by hops along adjacent cells on its S–D line. From

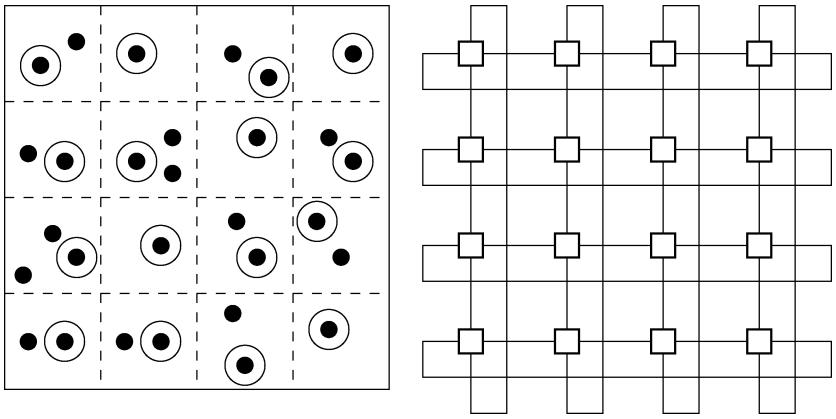

Fig. 3. The torus on the left with has 16 cells and each cell contains at least one node. The circled node in each cell acts as a relay. The corresponding queuing network of 16 servers, with each server corresponding to a cell in the wireless network, is shown on the right.

Lemma 2, it follows that each cell gets to transmit a packet every $1+c_1$ time slots, or equivalently, the cell throughput is $\Theta(1)$. The total traffic through each cell is that due to all the S–D lines passing through the cell, which is $O\left(n\sqrt{a(n)}\right)$ since Condition B is also satisfied. This suggests that

$$T(n) = \Theta\left(1/n\sqrt{a(n)}\right)$$

is achievable, if the average delay is finite.

Delay Analysis: Next we analyze the average packet delay in the wireless network for Scheme Π when Conditions A and B are satisfied, i.e., when policy Σ_n is used. Dividing the unit torus into square cells of area $a(n)$ results in $1/a(n)$ cells. One of the nodes in each cell maintains a buffer and acts as a relay for all the S–D lines passing through that cell. These relay nodes are the circled nodes in Fig. 3. The buffer in each cell corresponds to a queue and the cell itself corresponds to a server that can transmit one packet from this queue once in $1 + c_1$ time slots. This is because each cell becomes active once in $1 + c_1$ time slots as described earlier. Since Scheme Π restricts direct transmissions to be between adjacent cells, each cell can receive from or transmit to any four of its adjacent cells. This determines the connectivity between the servers so that the entire wireless network corresponds to a discrete-time queuing network of $1/a(n)$ servers, where each server is connected to four others as shown in Fig. 3.

Note that the TDM between cells is such that in the c_1 slots before each cell becomes active again each of its neighbors becomes active exactly once. Hence, we can ignore the effect of cells becoming active at regular intervals and instead consider a discrete-time network of queues $\mathcal{N}_D$ where D signifies the discrete time nature of this network. The actual delay in the wireless network would then be $1 + c_1$ times the delay in $\mathcal{N}_D$.

Queuing Network $\mathcal{N}_D$: The discrete-time queuing network $\mathcal{N}_D$ consists of $1/a(n)$ servers, each of which can service one packet from its queue in a time slot if it is not empty. Moreover, each server is connected to four others as explained earlier. In the wireless network, packets travel from their sources to their destinations by hops along adjacent cells on their S–D lines. Thus, the route of a packet depends on the S–D pair to which it belongs. This means that in $\mathcal{N}_D$ there are $n/2$ customer routes corresponding to the $n/2$ S–D pairs. Recall that packets arrive in the wireless network at the sources according to independent Poisson processes of rate $T(n)$. These correspond to exogenous arrivals at the queues in $\mathcal{N}_D$. The remaining arrivals at the queues are due to the departures from other queues. In the terminology of queuing theory, $\mathcal{N}_D$ is a discrete-time, open network of queues with general customer routes (see [9, Ch. 6.6]).

Delay analysis for such discrete-time networks with general customer routes is not known, which prevents us from using a simple FIFO order of service in $\mathcal{N}_D$. We leverage results known about continuous-time networks to obtain the queue management policy for $\mathcal{N}_D$ in such a way that the average delay can be computed.

Queuing Network $\mathcal{N}_C$: Consider a continuous-time open network of $1/a(n)$ servers having the same connectivity structure as $\mathcal{N}_D$ and the same $n/2$ customer routes (see Fig. 3). Let this network be called $\mathcal{N}_C$. Further, let the exogenous arrivals in both the networks $\mathcal{N}_C$ and $\mathcal{N}_D$ be the same. And let the service requirement of each packet at each server be deterministically equal to unit time. From the description until now, it is clear that $\mathcal{N}_C$ is the continuous-time analog of the discrete-time network $\mathcal{N}_D$. A PL queue management is used at each server in $\mathcal{N}_C$ (see [9, Ch. 6.8] for more details).

The queue size distribution for the continuous time network $\mathcal{N}_C$ with PL queue management at each server has a product form in equilibrium as shown in [5] (see Theorems 3.7 and 3.8 of Chapter 3) provided that the following two conditions are satisfied. First, the service time distribution should be either phase-type (that is, a mixture of Gamma distributions) or the limit of a sequence of phase-type distributions. The second condition is that, the total traffic at each server is less than its capacity, which is one in our case.

In our case, the service time is constant and equal to 1. Consider the sum of n exponential random variables each with mean $1/n$. This sum has a phase-type distribution and in the limit as n tends to infinity, its distribution converges to that of a constant random variable. Thus the first condition is satisfied.

In the wireless network, the number of S–D lines passing through each cell is $O(n\sqrt{a(n)})$ and the arrival process for each S–D pair is an independent Poisson process with rate $T(n) = \Theta(1/n\sqrt{a(n)})$. Therefore, an appropriate choice of constants guarantees that the total traffic at each server is less than 1, its service capacity, as Condition B (mentioned just before the description of policy Σ_n) is satisfied. Thus, the second condition is also satisfied.

Using the product form for the queue size distribution in equilibrium, it follows that the average queue size at a queue with total traffic $\lambda < 1$ and unit mean service is of the form $c_3\lambda/(1-\lambda)$ where c_3 is some constant. By Little's law, this implies that the average delay at each server is bounded above by a constant independent of n. We summarize the above discussion in the following lemma.

Lemma 4: For the continuous-time open network $\mathcal{N}_C$ with $n/2$ customer routes as described above the average delay at each server is bounded above by a constant independent of n.

Packet Scheduling in $\mathcal{N}_D$ Using $\mathcal{N}_C$: However, we cannot use this PL policy in the discrete time network $\mathcal{N}_D$ because of the following reasons.

1) Due to the discrete time nature of the network $\mathcal{N}_D$, a packet that is generated at time t becomes eligible for service (i.e., next hop transmission) only at time $\lceil t \rceil$.
2) A complete packet has to be transmitted in a time slot, i.e., fractions of the packets cannot be transmitted. This means that a preemptive type of service such as PL is not allowed.

To address these problems for $\mathcal{N}_D$, we present a centralized scheduling policy derived from emulating in parallel, the continuous-time network $\mathcal{N}_C$ with PL queue management at each server. The exogenous arrivals in both $\mathcal{N}_C$ and $\mathcal{N}_D$ are the same. Let a packet arrive in $\mathcal{N}_C$ at some server at time a_C and in $\mathcal{N}_D$ at the same server at time a_D. Then it is served in $\mathcal{N}_D$ using a LIFO policy with the arrival time considered to be $\lceil a_C \rceil$ instead of a_D.

Clearly such a scheduling policy can be implemented if and only if each packet arrives before its scheduled departure time. According to our scheduling policy, the scheduled departure time can be no earlier than $\lceil a_C \rceil$, whereas the actual arrival time is a_D. Hence for this scheduling policy to be feasible, it is sufficient to show that $a_D \leq \lceil a_C \rceil$ for every packet at each server. Let d_C and d_D be the departure times of a packet from some server in $\mathcal{N}_C$ and $\mathcal{N}_D$ respectively. Since the departure time at a server is the arrival time at the next server on the packet's route it is sufficient to show that $d_D \leq \lceil d_C \rceil$ for each packet in every busy cycle of each server in $\mathcal{N}_C$. In what follows, we show that for all packets in any busy cycle of any server, the departures in $\mathcal{N}_D$ occur at or before the departures in $\mathcal{N}_C$.

Lemma 5: Let a packet depart in $\mathcal{N}_C$ from some server at time d_C and in $\mathcal{N}_D$ at time d_D, then $d_D \leq \lceil d_C \rceil$.

Proof: Fix a server and a particular busy cycle of $\mathcal{N}_C$. Let it consist of packets numbered $1, \ldots, k$ with arrivals at times $a_1 \leq \cdots \leq a_k$ and departures at times $d_1, \ldots, d_k$. Let the arrival times of these packets in $\mathcal{N}_D$ be $A_1, \ldots, A_k$ and departures be at times $D_1, \ldots, D_k$. By assuming that $A_i \leq \lceil a_i \rceil$ for $i = 1, \ldots, k$, we need to show that $D_i \leq \lceil d_i \rceil$ for $i = 1, \ldots, k$.

Clearly, this holds for $k = 1$ since

$$D_1 = \lceil A_1 \rceil + 1 \leq \lceil a_i \rceil + 1 = \lceil d_1 \rceil.$$

Now suppose it holds for all busy cycles of length k and consider any busy cycle of $k+1$ packets.

If $\lceil a_1 \rceil < \lceil a_2 \rceil$, then because of the LIFO policy in $\mathcal{N}_D$ based on times a_i, we have

$$D_1 = \lceil a_1 \rceil + 1 \leq \lceil a_1 \rceil + k + 1 = \lceil d_1 \rceil.$$

The last equality holds since in $\mathcal{N}_C$, the PL service policy dictates that the first packet of the busy cycle is the last to depart. And the remaining packets would have departures times as for a busy cycle of length k.

Otherwise, if $\lceil a_1 \rceil = \lceil a_2 \rceil$, then the LIFO policy in $\mathcal{N}_D$ based on arrival times a_i results in

$$D_1 = \lceil a_1 \rceil + k + 1 = \lceil d_1 \rceil$$

and the packets numbered $2, \ldots, k$ depart exactly as if they belong to a busy cycle of length k. This completes the proof by induction. □

Thus we have shown that it is possible to use LIFO in $\mathcal{N}_D$ based on the arrival times in $\mathcal{N}_C$ instead of the actual arrival times in $\mathcal{N}_D$. We are now ready to prove Theorem 2.

Proof of Theorem 2 : Packets reach their destination with finite average delay, which shows that the throughput is just the rate at which each source sends its data. This proves that the throughput $T(n) = \Theta(1/n\sqrt{a(n)})$.

Next we compute the average packet delay $D(n)$. Lemma 5 also holds for the final departure of each packet from the network. Therefore, if D_D^i is the delay of a packet of route i in $\mathcal{N}_D$ (i.e., S–D pair i in the wireless network) and D_C^i is the delay of the corresponding packet in $\mathcal{N}_C$ then $D_D^i \leq D_C^i + 1$. Hence taking expectations it follows that

$$E[D_D^i] \leq E[D_C^i] + 1, \qquad 1 \leq i \leq n/2.$$

Therefore, delay averaged over all $n/2$ routes is given by

$$D(n) = \frac{2}{n} \sum_{i=1}^{n/2} E[D_D^i] \leq \frac{2}{n} \sum_{i=1}^{n/2} E[D_C^i] + 1. \tag{1}$$

Since each hop in the wireless network covers a distance of $\Theta(\sqrt{a(n)})$, the number of hops per packet for S–D pair i is $\Theta(d_i/\sqrt{a(n)})$, where d_i is the length of S–D line i. Now D_C^i is the delay for a packet of route i, which is equal to the sum of the delays along all queues on its route. But from Lemma 4, the average delay at each server is bounded above by some constant independent of n. Therefore, from (1), we obtain that

$$D(n) \leq \frac{2}{n} \sum_{i=1}^{n/2} c_2 \frac{E[d_i]}{\sqrt{a(n)}} + 1 = \Theta\left(1/\sqrt{a(n)}\right)$$

since

$$2 \sum_{i=1}^{n/2} E[d_i]/n = \Theta(1). \qquad \square$$

Finally, to see that the tradeoff provided by Scheme Π is optimal, consider the following result that was established for the fluid model in [2, Theorem 5].

Theorem 3: If any scheme has throughput$T(n)$ and delay $D(n)$, then $D(n) = \Omega(nT(n))$.

The constant packet size requirement is an additional constraint compared to the fluid model and hence its T–D cannot be better than that for the fluid model. This proves that the T–D scaling tradeoff provided by Scheme Π is optimal for the static random network model with packets of constant size.

III. Conclusion

The optimal T–D tradeoff for random wireless networks was determined in [1] with a more complete treatment in Part I [2] of this work. The analysis used a fluid model where the packet size needed to scale down with the number of nodes n in the network. In this correspondence, we imposed the constraint that the packet size remains constant and showed that the T–D tradeoff remains unchanged. This also provides a justification for the simplifying fluid assumption made in [1] and [2], since it does not affect the essential network dynamics.

The next natural question to address would be scaling in the mobile random network model with packets of constant size. In Part I, we showed that at throughput of $\Theta(1)$ (as in [3]), the optimal delay scaling is $\Theta(n \log n)$. Since the scheme used constant-size packets, this establishes the optimal delay scaling for the highest achievable throughput. The optimal tradeoff between throughput and delay for all lower throughputs, however, was achieved using a fluid model. The techniques developed in this correspondence cannot be applied directly to the mobile random network model due to the reason that nodes cannot be identified with cells as they are moving around the network. As a result, it is not possible to associate a virtual queue with each cell as we did in this work.

In a related model, where the mobile network also has n static nodes along with n mobile nodes, the optimal tradeoff can be obtained for sufficiently low throughputs. We can show that for any throughput $T(n) = \Theta(1/n^{1/2+\epsilon}), \epsilon > 0$, the tradeoff given by $T(n) = \Theta(D(n)/n)$ can be achieved. This is the same as the tradeoff for the fluid model in [2]. This establishes the optimal tradeoff for this range of low throughputs. The scheme achieving this tradeoff uses the scheduling scheme given in this correspondence along with a randomization technique and chasing in a manner similar to Scheme 3(a) in [2]. However, the optimal tradeoff for the mobile network with no static nodes is unknown.

REFERENCES

[1] A. El Gamal, J. Mammen, B. Prabhakar, and D. Shah, "Throughput-delay tradeoff in wireless networks," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004.

[2] ——, "Optimal throughput-delay scaling in wireless networks—Part I: The fluid model," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2568–2592, Jun. 2006.

[3] M. Grossglauser and D. M. C. Tse, "Mobility increases the capacity of ad-hoc wireless networks," in *Proc. IEEE INFOCOM*, Anchorage, AK, 2001, pp. 1360–1369.

[4] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[5] F. P. Kelly, *Reversibility and Stochastic Networks*. New York: Wiley, 1979.

[6] S. R. Kulkarni and P. Viswanath, "A deterministic approach to throughput scaling in wireless networks," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1041–1049, Jun. 2004.

[7] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[8] M. J. Neely and E. Modiano, "Capacity and delay tradeoffs for *ad hoc* mobile networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1917–1937, Jun. 2005.

[9] R. W. Wolff, *Stochastic Modeling and the Theory of Queues*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

Coding in the Block-Erasure Channel

Albert Guillén i Fàbregas, *Member, IEEE*

***Abstract*—In this correspondence, we study an M-ary block-erasure channel with B blocks, where with probability ϵ a block of L coded symbols is erased. The behavior of the error probability of coded systems over such channels is studied, and we show that, if the code is diversity-wise maximum-distance separable, its word error probability is equal to the outage probability, which admits a very simple expression. This correspondence is intended to complement the error probability analysis in previous work by Lapidoth and shed some light on the design of coding schemes for nonergodic channels.**

***Index Terms*—Diversity, erasure channels, error probability, maximum-distance separable (MDS) codes, maximum-likelihood (ML) decoding, nonergodic channels, outage probability.**

I. INTRODUCTION

The block-erasure channel is a very simplified model of a fading channel where parts of the codeword are completely erased by a deep fade of the channel [1]. This channel corresponds to the large signal-to-noise ratio (SNR) regime of the block-fading channel [2]–[7], and its interest lies on its simplicity and nonergodicity, typical of many real wireless communication systems, such as orthogonal frequency division multiplexing (OFDM) or frequency-hopped systems. Coding for the block-erasure channel with convolutional codes has been studied in some detail in [1]. In this context, non-ergodicity means that the transmitted codeword spans only a finite number B of independent realizations (degrees of freedom) of the channel irrespectively of its length.

In this correspondence we study the problem of *fixed-rate* transmission over the block-erasure channel. This correspondence complements previous error probability analysis for convolutional codes in the block-erasure channel done by Lapidoth in [1]. In particular, we derive simple expressions for the word and bit error probabilities of general codes of a fixed rate, as well as tight bounds. We find that diversity-wise maximum-distance separable (MDS) codes have the lowest possible error probability and are therefore optimal for this channel.

II. CHANNEL MODEL

We study a block-erasure channel with B blocks. With probability ϵ a block of L symbols is completely erased and with probability $1-\epsilon$ a block of L coded symbols is received correctly (noiseless sub-channel), independently from block to block. Consider the transmission of an M-ary code $\mathcal{C}$ of length $N = BL$ and rate $R = \frac{K}{N}$ bits per channel use, where $K = \log_2 |\mathcal{C}|$. Also, let $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_B) \in \{0, 1, \ldots, M-1\}^N$ be a codeword of $\mathcal{C}$. We denote erasures by "?." The block-erasure channel is illustrated in Fig. 1.

Manuscript received September 23, 2005; revised August 22. 2006. This work has been supported by the Australian Research Council (ARC) under Grant DP0558861 and by the University of South Australia Australian Competitive Grant Development Scheme. The material in this correspondence was presented in part at the 43rd Annual Allerton Conference on Communication, Control and Computing, Monticello, IL, September 2005 and at the 2006 Australian Communications Theory Workshop, Perth, Australia, February 2006.

The author is with the Institute for Telecommunications Research, University of South Australia, SPRI Building, Mawson Lakes, SA 5095, Australia (e-mail: guillen@ieee.org).

Communicated by M. P. Fossorier, Associate Editor for Coding Techniques.

Color version of Fig. 3 is available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2006.883556