Asymptotically Optimal Data Dissemination in Multi-Channel Wireless Sensor Networks: Single Radios Suffice

David Starobinski* Weiyao Xiao*
* Department of Electrical and Computer Engineering
Boston University, Boston, MA 02215
Email:{staro,weiyao}@bu.edu

***Abstract*—We analyze the performance limits of data dissemination with multi-channel, single radio sensors under random packet loss. We formulate the problem of minimizing the average delay of data dissemination as a stochastic shortest path problem and show that, for an arbitrary topology network, an optimal control policy can be found in a finite number of steps, using value iteration or Dijkstra's algorithm. However, the computational complexity of this solution is generally prohibitive. We thus focus on two special classes of network topologies of practical interest, namely single-hop clusters and multi-hop cluster chains. For these topologies, we derive the structure of policies that achieve an asymptotically optimal average delay, in networks with large number of nodes. Our analysis reveals that a single radio in each node suffices to achieve performance gain directly proportional to the total number of channels available. Through simulation, we show that the derived policies perform close to optimal even for networks with small and moderate numbers of nodes and can be implemented with limited overhead.**

***Index Terms*—Broadcast, Over-the-Air Programming, Delay Minimization, Scheduling, Stochastic Shortest Path, Extreme Value Theory.**

I. Introduction

Recent technological improvements in micro-electro-mechanical systems (MEMS) have made the deployment of large scale wireless sensor networks a reality. Wireless sensor networks usually consist of many small size, inexpensive, distributed sensor nodes. These sensor nodes are able to sense, compute and communicate wirelessly.

A wide variety of fundamental sensor networking services, such as routing and over-the-air programming, rely upon efficient data dissemination [2]–[4]. However, since sensor networks tend to be dense, data dissemination without careful management can easily cause significant problems, such as the well-known "broadcast storm" [5] problem. Moreover, traditional sensor limitations (e.g., limited battery life and memory) joined with the complications of the wireless sensor broadcast channel (e.g., lossy channel, narrow bands, and energy-expensive communication) make it extremely difficult to implement efficient dissemination algorithms.

Fortunately, wireless sensor radios currently on the market do enjoy at least one currently under-utilized feature: they are able to communicate on any one of multiple (narrow) channels [6]–[10]. Thus, for example, MICA2 sensor motes operating in the 900 Mhz range can communicate on any one of more than 25 non-overlapping channels. Ideally, network traffic would be evenly split onto all the channels available, leading to a drastic improvement in the efficiency and scalability of data dissemination in sensor networks.

The main challenge, however, is that sensor nodes are equipped with a single radio interface and, thus, can operate on only one of these channels at a time. The main effort of this work is to demonstrate, through theoretical analysis and simulation, that the multi-channel transceiving capability of sensor motes can nevertheless be exploited for major efficiency gains.

Specifically, we propose a theoretical framework to evaluate the performance limits of data dissemination with multi-channel, single radio sensors, using expected delay (completion time) with respect to the probability distribution of packet loss as the primary optimization metric. Within this framework, we show how to model the problem of data dissemination as an instance of the stochastic shortest path problem [11]. This framework permits us to find an optimal dissemination policy for an arbitrary topology in a finite amount of time using value iteration or Dijkstra's algorithm. However, the computational complexity for deriving the optimal policy is generally prohibitive as it grows exponentially with the system parameters.

The optimal solutions are typically very complicated and nonintuitive. As such, we focus on two specific classes of topologies of practical interest: single-hop clusters and multi-hop cluster chains [12], [13]. For large size networks with these topologies, we derive the structure of policies that exhibit asymptotically optimal expected delay, in networks with large number of nodes. These policies make use of a round robin strategy applied both at the packet and channel levels. We, thus, refer to them as packet-channel round robin (PCRR) policies. One of our main theoretical contributions is to show that, with C channels available, the expected delay achieved with PCRR approaches a value that is C times smaller than the optimal expected delay with a single channel policy. Thus, a single radio interface in each node suffices to achieve performance gain proportional to the total number of channels available. Our results are validated by simulation, showing

A preliminary version of this paper appeared in the proceedings of the IEEE INFOCOM 2007 conference [1].

This research was supported in part by the US National Science Foundation under grants CNS-0435312, CCF-0729158, and CCF-0916892.

that PCRR policies are nearly optimal even for small and moderately-sized networks.

Our work provides a first step in rigorously characterizing the performance limits of multi-channel, single radio wireless sensor networks. Although control overhead and channel switching latencies are not explicitly captured in the theoretical model, we show that the PCRR policy is implementable in practice and propose a protocol design to this effect. We evaluate the performance of the protocol with TOSSIM, a bit-level simulator designed specifically for TinyOS-based sensor networks [14]. The simulations show that as few as two channels suffice for PCRR to widely outperform a baseline data dissemination protocol operating on a single channel.

The rest of this paper is organized as follows. In Section II, we describe work related to our research. In Section III, we first present our model. Next, we formalize our optimization problem, analyze its computational complexity, and show how it can (theoretically) be solved in finite time. In Section IV, we analyze the problem of data dissemination in single cluster topologies. We introduce the PCRR policy and prove its asymptotical optimality in networks with large number of nodes. In Section V, we generalize our results to multi-hop cluster chains topologies. Numerical results are presented in Section VI. We describe and evaluate a practical implementation of the PCRR policy in Section VII and provide concluding remarks in Section VIII.

II. Related Work

Data dissemination has many applications in wireless sensor networks. Over-the-Air Programming (OAP) is one of the most important. Considerable research has been conducted on designing low-delay OAP protocols for disseminating a new program to an entire network, see, e.g., Deluge [2] and MNP [4]. These protocols, however, communicate only over a single channel. Recently, a number of contributions have suggested to exploit the multi-channel resources available in sensor nodes to speed-up OAP [6]–[8]. However, none of them provide any form of theoretical guarantees on the completion time of data dissemination.

Several MAC and routing protocols for wireless nodes equipped with multi-channel single radios have been proposed in the literature, e.g., in [15]–[19]. However they are not directly applicable to our setting because they focus on unicast communication and do not aim at minimizing the completion time of data dissemination.

The work in [20] analyzes the delay performance of a multi-channel MAC protocol in a point-to-multipoint wireless network. Contrary to our setting, this paper assumes that the base station can communicate over multiple channels simultaneously and that each channel is used for unicast communication.

Reference [21] conducts an analysis on the *capacity* of multi-channel wireless networks. The paper studies how the number of radios and channels available in each node affects the capacity of the network. Our work, on the other hand, focuses on the *delay* performance of broadcasting data packets from one or a few sources to the whole network using multi-channel single radio nodes.

A number of papers have studied the problem of optimizing multi-channel broadcast schedules in single-hop networks, see, e.g., [22], [23]. The model considered in these papers is different from ours. It assumes that clients continuously request items stored at the base station. The goal is to determine the optimal schedule of transmissions by the base station, based on the degree of popularity of the items, such that the average access time (i.e., the time between a request and its response) is minimized.

The problem of minimizing the delay of file dissemination in single chain and multi-chain wireless network topologies is considered in [24]. The multi-hop model of that paper differs from ours because it assume that links are lossless and that only one node is present at each hop. That said, the scheduling policies of [24] form the basis of the first stage of the multi-hop PCRR policy described in Section V and in the Appendix.

Reference [25] analyzes the benefit of exploiting multi-channel communication for reliable multicast in wireline networks. The paper derives a numerical expression for the expected number of transmissions needed for all the receivers to receive a single packet correctly. In our paper, we consider the case where several packets must be received by all the nodes and provide explicit, analytical expressions on the expected delay, when the number of receivers is large.

Round-robin policies resembling the PCRR policy have been studied in high-speed and optical networks [26]–[29]. One of the main contributions of our work is to provide stochastic bounds on the delay performance of the PCRR policy and prove its asymptotical optimality within the context of data dissemination in sensor networks.

III. Model and Problem Formulation

A. Model

We consider the problem of disseminating a file consisting of M packets from a set of S sources (e.g., base stations) to N nodes in an arbitrary topology network, with C orthogonal channels available for communication. Each source has a copy of the entire file. The time axis is slotted and each packet transmission takes one time slot. Each node is equipped with a single, half-duplex radio. Thus, during a time slot, a node can either transmit or receive (but not both) on one of the C channels. To simplify exposition, we assume that packets do not need to be received in order at the various nodes for a file to be properly reconstructed although our results hold without this assumption (see discussion at the end of Section IV). Note that several data dissemination protocols in wireless sensor networks, such as Deluge [2], do not require that packets be received in order.

At each time slot, a control (action) u specifies for each node whether it transmits or not and the channel to which it tunes. Packets are not only transmitted by sources but possibly also by other nodes that have received some of the packets and serve as relays. Communications take place over a wireless broadcast channel, whose losses are independent and identically distributed at each time slot. As such, we can associate a probability $p_{ij}(u)$ of a packet transmission from node i to j being corrupted; this probability is a function of the control u because the packet loss is dependent on all simultaneous transmissions on the same channel.

Finally, we will denote by T the random variable representing the time (delay) until all nodes receive all packets. Our goal is thus to determine a control policy that minimizes the expected value of T (denoted $\bar{T}$).

B. Problem Formulation for General Networks

We next formalize our optimization problem and provide a computational methodology to solve it by casting it as a stochastic shortest path (SSP) problem [11]. In our specific case, this problem can be solved deterministically in bounded time using value iteration or Dijkstra's algorithm [30].

a) SSP problem: The SSP problem is a generalization of the deterministic shortest path problem in a graph. Specifically, in the stochastic version, a path from a source to destination is determined probabilistically, meaning that one may transition from a vertex to any other vertex according to a given distribution (which, in turn, is determined by a chosen control). Each vertex corresponds to a different state. The shortest path, in this context, corresponds to the choice of controls at each vertex that minimize the expected cost to a given destination (or *termination state* in the literature). Clearly, the deterministic version of the problem thus corresponds to a case in which, controls from any given vertex assign a probability 1 for reaching some vertex and 0 for reaching all others.

b) Formulation: In our case, we build a graph of $|V| = 2^{NM}$ vertices, each of which correspond to an $N \times M$ binary matrix representing a possible configuration of the network in the middle of a data dissemination process. Specifically, the (n, m)-th entry of any such matrix is 1 if and only if node n has received packet m in the corresponding configuration. For simplicity, we order the states so that the initial state $i = 1$ and last state $i = |V|$ correspond to the all zero and all one matrices respectively.

To complete the model, we assume a set of possible controls $U(i)$ for each state i, and define a corresponding transition probability $q_{ij}(u)$ corresponding to the probability of reaching state j from state i if transmissions are enacted according to control $u \in U(i)$. Our goal is then to determine an optimal control $\pi(i) \in U(i)$ at every state so as to minimize the expected delay from state 1 to the termination state $|V|$. The optimal control policy π is guaranteed to be stationary because the channel is assumed to be $i.i.d.$

c) Solution: We use the random variable $T^*_{DP}(i)$ to denote the time to reach the termination state starting from state i, using the optimal policy. Its expected value, $\bar{T}^*_{DP}(i)$, is solved using dynamic programming. One of the main results for the SSP problem is that it has a unique solution satisfying Bellman's equations [11], [31]:

$$\bar{T}^*_{DP}(i) = \min_{u \in U(i)} \left[1 + \sum_{j=1}^{|V|} q_{ij}(u) \bar{T}^*_{DP}(j) \right], i = 1, \ldots, |V|. \quad (1)$$

For each state i, the optimal control $\pi(i)$ corresponds to the argument that achieves the minimum in the right-hand side of Eq. 1. To simplify notation, we will use $\bar{T}^*$ to represent the optimal expected delay starting from initial state 1. Traditional approaches to this solution include *value iteration*, the most commonly used approach that generally requires an infinite number of iterations for convergence, and *policy iteration*, which is more computationally expensive at each step but terminates in finite time [11].

The special structure of our problem allows for an especially efficient value iteration solution satisfying (1). Specifically, our graph has acyclic transition probabilities, in that a path can never visit the same state twice, except for self-transitions (which can be eliminated [11, Vol. 2: p. 97]); this is because nodes cannot lose packets that they correctly received and decoded. As such, value iteration is guaranteed to converge within $|V|$ iterations, where V is the state space. Since each iteration involves $|V|^2|U|$ operations, where $|U|$ is the size of the control space, the optimal solution can be computed with at most $|V|^3|U|$ operations. The following theorem summarizes this result.

Theorem 1: The optimal dissemination of an M-packet file to a network with N nodes, S sources, and C channels can be computed using value iteration with at most $|V|^3|U|$ operations, where $|V| = 2^{NM}$ and $|U| = 2^{(N+S)C}$.

We further note that the optimal policy must be *consistently improving*, meaning that

$$q_{ij}(\pi(i)) > 0 \quad \Rightarrow \quad \bar{T}^*_{DP}(i) > \bar{T}^*_{DP}(j).$$

We can then use Dijkstra's algorithm or any label-based shortest path algorithms to compute the optimal policy and optimal expected delays [11, Vol. 2: p. 135]. However, the worst-case computational complexity of the problem remains the same as stated by Theorem 1.

C. Problem Formulation for Single Clusters and Cluster Chains

As discussed above, in general, the derivation of an optimal policy for our problem is computationally prohibitive for large numbers of nodes, packets, or channels. As such, the remainder of this work focuses on analytical approaches for some specific topologies of practical interest, such as single-hop clusters and (multi-hop) cluster chains. Even in such cases, the optimal policies can be quite involved, but we are able to derive asymptotically optimal policies, as $N \to \infty$.

1) Single Clusters: In this model, the network consists of a a single-hop topology (shown in Figure 1), where every node is within one-hop communication range of every other node. On each channel, to avoid packets collision, only one node can transmit (broadcast) in each time slot. For each receiver, the packet loss probability is p, independently of any other events (e.g., reception of the packet by other nodes).

2) Cluster Chains: This model corresponds to a collection of clusters organized along W chains (see Figure 2). It is a special case of tree structure, where each cluster has at most one child cluster. Similar topologies have previously been considered in the literature. For instance, the PEGASIS protocol [13] arranges sensor nodes along a chain to achieve energy-efficient data aggregation in a sensor network.

We index the chains using the variable w, where $1 \leq w \leq W$. We use the tuple (w, d) to refer to a cluster along the

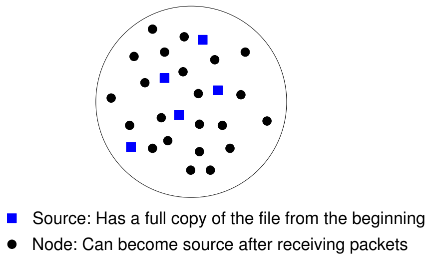


Fig. 1. Single cluster topology.

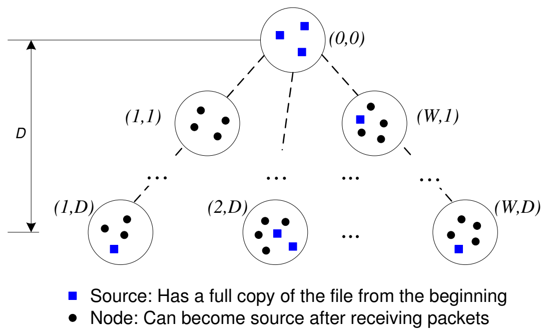


Fig. 2. Cluster chains topology.

w-th chain that is d hops away from the root cluster $(0,0)$. Define $D(w)$ to be the length (in terms of number of hops) of chain w and $D = \max_{1\leq w\leq W} D(w)$ be the length of the longest chain. For the purpose of our asymptotic analysis, we assume that the number of nodes in each cluster (w,d) scales linearly with the total number of nodes in the network, i.e., $N_{wd} = \alpha_{wd}N$, where $\sum_{w=1}^{W}\sum_{d=1}^{D}\alpha_{wd} = 1$. Note that if $d > D(w)$ then, by definition, $\alpha_{wd} = 0$.

We assume that the root cluster contains one or more sources. All other clusters may or may not contain sources. The total number of sources is S. A node in a cluster can communicate directly with all the other nodes in the same cluster as well as with nodes belonging to a direct parent or child cluster. We assume a 2-hop interference model. Thus, if a node transmits on a certain channel, then all the nodes belonging to any ancestor or descendant clusters within a distance of two hops and operating on the same channel must remain quiet to avoid packet collision. We also assume that clusters belonging to different chains do not interfere with each other. As earlier, the packet loss probability is fixed to p for each pair of source and destination. In this paper, we do not enter into detail on how to decompose a network into clusters. We refer the interested reader to [32]–[34] for possible approaches.

IV. Analysis of Single Clusters

In this section, we analyze the single cluster model described in Section III-C. We first consider the single channel case. We determine optimal policy and derive tight bounds on the optimal expected delay. We then consider the multi-channel case. We first derive a lower bound on the optimal expected delay and then determine a policy that is asymptotically optimal as $N \to \infty$, while keeping all the other parameters constant. These results are first proven for the case when the number of sources is larger or equal to the number of channels, i.e., $S \geq C$, and then extended to the case $S < C$. In the following, we denote by $T_N^*(C)$ the random variable representing the completion time using the optimal policy in a cluster of N nodes with C channels available for communication. Our model implicitly assumes the existence of a control channel through which a source is notified when all the receivers have received a given packet. The impact of this control overhead is evaluated in VII, where we describe a practical protocol implementation of the PCRR policy together with TOSSIM simulations.

Before proceeding with the analysis, we first recall two standard asymptotic notations that will be used to characterize the asymptotic growth of the expected delay [30]:

- Θ-notation: For two given functions $f(n)$ and $g(n)$, we say that $f(n) = \Theta(g(n))$ if there exist positive constants c_1, c_2, and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$.
- o-notation: For given functions $f(n)$ and $g(n)$, we say that $f(n) = o(g(n))$ if for any positive constant ϵ there exists a constant $n_0(\epsilon)$ such that $0 \leq f(n) < \epsilon g(n)$ for all $n \geq n_0(\epsilon)$.

A. Single Channel

With a single channel, only one source can transmit in any time slot. Denote by T_n^m the number of slots needed for user n to receive packet m. Under the assumption of a packet dropping probability p, T_n^m has a geometric distribution with mean $1/(1-p)$. Thus, because of the broadcast nature of the channel, the number of slots needed for all the users to receive packet m is $T^m = \max_{n=1,\ldots,N} T_n^m$. The minimum number of time slots to complete the transmission of all the packets is thus $T_N^*(1) = \sum_{m=1}^{M} T^m$. The order in which the source transmits the packets is arbitrary. Thus, any transmission policy in which, in each time slot, the source transmits a packet needed by at least one of the N nodes in the network is optimal.

We next provide an asymptotic expression for the optimal expected delay in the case where only a single channel is available.

Proposition 1: Consider a single cluster topology with $C = 1$ channel. Then, as $N \to \infty$,

$$\bar{T}_N^*(1) = M\lambda^{-1}\log(N) + \Theta(1),$$

where $\lambda = \log(1/p)$.

Proof: For the random variables T_n^m, we have $\Pr\{T_n^m = i\} = p^{i-1}(1-p)$ and $\Pr\{T_n^m \geq i\} = \sum_{j=i}^{\infty} p^{j-1}(1-p) = p^{i-1}$, $i \geq 1$. Consider an equivalent continuous random variable X_n^m, where X_n^m has the pdf $f_{X_n^m}(x) = \sum_{i=1}^{\infty} p^{i-1}(1-p)\delta(x-i)$, where $\delta(x)$ is the Dirac's delta function. The ccdf of X_n^m is $\bar{F}_{X_n^m}(x) = p^{\lfloor x \rfloor}$, for $x \geq 0$.

We now define the random variables Y_n^m and $Z_n^m = Y_n^m + 1$ where $\bar{F}_{Y_n^m}(x) = p^x$ and $\bar{F}_{Z_n^m}(x) = \min(1, p^{x-1})$. We note that Y_n^m are independent exponential random variables with parameter $\lambda = \log(\frac{1}{p})$. Clearly, we have $\bar{F}_{Y_n^m}(x) \leq \bar{F}_{X_n^m}(x) \leq \bar{F}_{Z_n^m}(x)$. Therefore, $Y_n^m \leq_{st} X_n^m \leq_{st} Z_n^m$, where the notation $Y \leq_{st} X$ means that the random variable Y is stochastically smaller than the random variable X [35, p. 404]. Using properties of stochastic ordering, we then obtain

$$\mathbb{E} \max_{n=1,\dots,N} Y_n^m \leq \mathbb{E} \max_{n=1,\dots,N} X_n^m \leq \mathbb{E} \max_{n=1,\dots,N} Z_n^m. \quad (2)$$

Since Y_n^m are independent exponential random variables, from [36, p. 73], we have

$$\mathbb{E} \max_{n=1,\dots,N} Y_n^m = \sum_{n=1}^{N} \frac{1}{n\lambda}. \quad (3)$$

Using known bounds on the harmonic sum [37], as $N \to \infty$, we have,

$$\sum_{n=1}^{N} \frac{1}{n} \leq \log(N) + \frac{1}{2N} + \gamma \quad (4)$$

and

$$\sum_{n=1}^{N} \frac{1}{n} \geq \log(N) + \frac{1}{2(N+1)} + \gamma, \quad (5)$$

where $\gamma \approx 0.577215$ is the Euler's constant.

Since $Z_n^m = Y_n^m + 1$, from Eqs. (2), (4) and (5), we obtain

$$\mathbb{E} \max_{n=1,\dots,N} X_n^m = \lambda^{-1} \log(N) + \Theta(1). \quad (6)$$

Since $\mathbb{E}T^m = \mathbb{E} \max_{n=1,\dots,N} X_n^m$, we deduce from Eq. (6)

$$\begin{aligned} \bar{T}_N^*(1) &= M\mathbb{E}T^m \\ &= M\lambda^{-1} \log(N) + M\Theta(1) \quad (7) \\ &= M\lambda^{-1} \log(N) + \Theta(1). \quad (8) \end{aligned}$$

The transition from Eq. (7) to Eq. (8) is justified by the fact that $M = \Theta(1)$, since M is kept constant while $N \to \infty$. ■

B. Multiple Channels

We now address the case where $C > 1$. In general, the structure of the optimal policy appears to be quite intricate. One exception is the unconstrained channel case, that is, $C \geq M$, in which a distinct channel can be dedicated to the transmission of each packet. Note that when the number of channels C is greater or equal to the number of packets M, then $T_N^*(C) = T_N^*(M)$.

We will exploit the results obtained for the single channel and unconstrained channel cases to provide bounds on the optimal expected delay for the case $1 < C < M$. Using these bounds, we will show that, as $N \to \infty$, a simple scheduling policy, called Packet-Channel Round-Robin, achieves an asymptotically optimal expected delay. The results in this section are first derived for the case when the number of sources is larger or equal to the number of channels, i.e., $S \geq C$, or $S \geq M$ in the case where $C \geq M$ since the system never needs to use more than M channels. At the end of the section, we show how they can be easily extended to the case $S < C$.

a) Unconstrained channels: Suppose $C \geq M$. Then, a different packet can be continuously transmitted on each channel, e.g., packet 1 on channel 1, packet 2 on channel 2, etc. As in the single channel case, the number of slots needed for user n to receive packet m, denoted by T_n^m, is geometrically distributed with parameter p. However, this time, as soon as user n receives packet m, it can switch to another channel to receive the next packet it needs, and so on until all the packets are received. The time needed for user n to receive all the packets is, thus, $T_n = \sum_{m=1}^{M} T_n^m$ and the completion time for all the nodes is $T_N^*(M) = \max_{n=1,\dots,N} T_n$. Any control policy is optimal as long as nodes tune to channels on which they can receive a new packet.

We next provide an expression for the optimal expected delay. First we note that being the sum of independent geometric random variables, T_n is a Pascal (or negative binomial) random variable with parameters p and M. We can then use a procedure similar to the proof of Theorem 1. We replace T_n by an equivalent continuous random variable and stochastically bound it. We then obtain the following result:

Proposition 2: Consider a single cluster topology with $C \geq M$ channels and $S \geq M$ sources. Then, as $N \to \infty$,

$$\bar{T}_N^*(M) = \lambda^{-1}[\log(N) + (M-1)\log\log(N)] + \Theta(1),$$

where $\lambda = \log(1/p)$.

Proof: Similar to the proof in Proposition 1, we use the same random variables X_n^m, Y_n^m and Z_n^m. Let $X_n = \sum_{m=1}^{M} X_n^m$, $Y_n = \sum_{m=1}^{M} Y_n^m$ and $Z_n = \sum_{m=1}^{M} Z_n^m$. Using stochastic ordering properties, we have

$$\bar{F}_{Y_n}(x) \leq \bar{F}_{X_n}(x) \leq \bar{F}_{Z_n}(x).$$

Therefore, $\mathbb{E} \max_{i=1,\dots,N} Y_i \leq \bar{T}_N^*(M) \leq \mathbb{E} \max_{i=1,\dots,N} Z_i$. As the sum of M i.i.d. exponential random variables, Y_n is an Erlang random variable. From extreme value theory results on the convergence in probability of the maximum of independent Erlang random variables [38] and result on moments convergence [39], we have, as $N \to \infty$,

$$\mathbb{E} \max_{i=1,\dots,N} Y_i = \lambda^{-1}[\log(N) + (M-1)\log\log(N)] + \Theta(1).$$

Since $\mathbb{E} \max_{i=1,\dots,N} Z_i = \mathbb{E} \max_{i=1,\dots,N} Y_i + M$, the result follows. ■

b) Constrained channels: Assume now $C \leq M$. We next show that, for the optimal control policies, the completion time with C channels is always stochastically larger than the completion time in a single channel system running C times faster.

Theorem 2: Consider a single-hop cluster with $C \leq M$ channels. Then $T_N^*(C) \geq_{st} \frac{1}{C} T_N^*(1)$.

Proof: We prove this result using a sample path argument (coupling) [35, p. 409]. Consider an arbitrary time slot in the system with C channels, denoted by SYS_C, assuming packet m_i is transmitted in channel i, $i = 1, .., C$. Denote an equivalent single channel system by SYS_1 evolving over C time slots. Each event in SYS_C is mapped to SYS_1 by having packet m_i being sent in time slot i, $i = 1, .., C$. Since we have single radio, a node in SYS_C can listen on one channel only, says channel j. In SYS_1, the same node listens only during slot j but is forced not to listen during all the other slots. Clearly, given the same starting state, SYS_1 using C

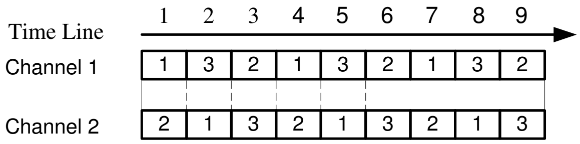


Fig. 3. Illustration of the PCRR policy for $M = 3$ packets and $C = 2$ channels. The number shown in each block corresponds to the index of the packet being transmitted.

time slots is equivalent to SYS_C using one time slot. The optimal policy using a single channel will always perform at least as well as SYS_1, since, in practice, nodes are allowed to listen to packet transmissions in each time slot. Therefore, $CT^*_N(C) \geq_{st} T^*_N(1)$, and the theorem follows. ■

We now introduce a simple control policy called Packet-Channel Round-Robin (PCRR). We will show that this policy is asymptotically optimal, as $N \to \infty$. To explain the policy, we introduce a few notations. We use the variable c to index channels, i.e., $c = 1, 2 \ldots, C$, the variable m to index packets, i.e., $m = 1, \ldots, M$, and the variable t to index time slots, i.e., $t = 1, 2, \ldots, T_{PCRR}$. The random variable T_{PCRR} represents the final time slot, that is, the time slot at which all the nodes have completed receiving all the packets. The PCRR transmission policy states that at time slot t, packet $[(C(t-1) + c - 1) \mod (M) + 1]$ should be transmitted on channel c. A pseudo-code for the algorithm is shown in Algorithm 1. An illustration of the policy for the case $C = 2$ and $M = 3$ is given in Fig. 3.

The reception policy of PCRR is as follows. At each time slot, a destination node should select a channel on which a missing packet is being transmitted. If multiple missing packets are transmitted concurrently on different channels, then a node should listen to the channel on which the packet with the smallest index number is being transmitted.

Algorithm 1 Packet Channel Round Robin: PCRR(M,C)

Input:
Number of packets need to be disseminated, M;
Number of channels available, C;
Output:
Completion time, T_{PCRR};

```
t ← 1;
repeat
    for c = 1 to C do
        Transmit packet m_c(t) on channel c at time t,
        where
        m_c(t) = [(C(t−1) + c − 1) mod (M) + 1];
    end for
    t = t + 1;
until all the nodes have received all the packets
return T_PCRR = t;
```

We next provide a stochastic relation between the completion time of the PCRR policy using C channels, denoted by $T_{PCRR}(C)$, and the completion time in the unconstrained channel case using the optimal policy, $T^*_N(M)$.

Theorem 3: Consider a single-hop cluster with $C \leq M$ channels. Then $T_{PCRR}(C) \leq_{st} \frac{M}{C} T^*_N(M) + 1$.

Proof: The proof is again based on a sample path argument. Consider an unconstrained system with M channels, denoted by SYS_M. Without loss of generality, assume packet i is always transmitted on channel i and nodes receive packets in increasing order, namely, a node starts trying to receive packet $(i+1)$ only if it has already received packets $1, 2, \ldots, i$. As earlier, we use the notation T_n to represent the completion time of node n.

Denote by SYS'_{PCRR}, a system with C channels implementing the PCRR transmission policy, but with some additional restrictions on the reception policy of the nodes as detailed below. Thus, if in SYS_M packet i is transmitted at time t, then in SYS'_{PCRR}, the same packet is transmitted on channel $\left([(t-1)M - C\lfloor\frac{(t-1)M}{C}\rfloor + i - 1] \mod C + 1\right)$ at time $\lceil\frac{(t-1)M+i}{C}\rceil$, $i = 1, .., M$, in accordance with the packet-channel round-robin schedule.

We next describe the reception policy of SYS'_{PCRR}. Consider an arbitrary realization in SYS_M and suppose that node n listens to packet $m_n(t)$ during time slot t, where $n = 1, .., N$, $t = 1, .., T_n$, and $m_n(t) \in [1, M]$. In SYS'_{PCRR}, this event is mapped by having node n listen to packet $m_n(t)$ during time slot $\lceil\frac{(t-1)M+m_n(t)}{C}\rceil$ on channel $\left([(t-1)M - C\lfloor\frac{(t-1)M}{C}\rfloor + m_n(t) - 1] \mod C + 1\right)$. In addition, SYS'_{PCRR} prohibits node n to listen to packets transmissions during other time slots that do not follow this mapping.

We next prove the feasibility of the above mapping by showing that each node in SYS'_{PCRR} is required to listen to at most one packet during every time slot. Suppose in SYS_M, node n listens, respectively, to packet $m_n(t)$ and $m_n(t+1)$ at time t and $t+1$. Correspondingly, in SYS'_{PCRR}, node n listens to packets $m_n(t)$ and $m_n(t+1)$ at time $\lceil\frac{(t-1)M+m_n(t)}{C}\rceil$ and $\lceil\frac{tM+m_n(t+1)}{C}\rceil$, respectively. Since nodes receive packets in increasing order in SYS_M, we know that $m_n(t+1) \geq m_n(t)$. Furthermore, since $C \leq M$, we have $\lceil\frac{tM+m_n(t+1)}{C}\rceil > \lceil\frac{(t-1)M+m_n(t)}{C}\rceil$, which means that node n is required to listen to at most one packet during any time slot. Thus, SYS'_{PCRR} using $\lceil\frac{TM}{C}\rceil$ time slots is equivalent to SYS_M using T time slots.

Now consider system SYS_{PCRR} employing the original PCRR policy without forcing nodes not to listen packets during certain time slots. We recall that nodes attempt to receive missing packets in increasing order. Thus, if node n has not received packet $m_n(t)$ by time slot $\lceil\frac{(t-1)M+m_n(t)}{C}\rceil$, it will listen to the transmission of this packet in that slot, that is, it behaves the same as it would in SYS'_{PCRR}. Otherwise, if it did receive packet $m_n(t)$ before time slot $\lceil\frac{(t-1)M+m_n(t)}{C}\rceil \leq \lceil\frac{tM+m_n(t)}{C}\rceil$, then it definitely would have received all the packets up to packet $m_n(t)$ before SYS'_{PCRR}. Therefore, for any sample path in SYS_M completing in T time slots, SYS_{PCRR} is guaranteed to complete its sample path in no more than $\lceil\frac{TM}{C}\rceil$ slots and the theorem follows. ■

We now prove the main result of this section, namely that the PCRR policy is asymptotically optimal, as $N \to \infty$.

Theorem 4: Consider a single-hop cluster with $C \leq M$ channels. Then

$$\lim_{N\to\infty} \frac{\bar{T}_{PCRR}(C)}{\bar{T}^*_N(C)} = 1.$$

Proof: From Theorems 2 and 3, we have

$$\frac{1}{C}T^*_N(1) \leq_{st} T^*_N(C) \leq_{st} T_{PCRR}(C) \leq_{st} \frac{M}{C}T^*_N(M) + 1.$$

From stochastic ordering properties, similar inequalities apply for the expectations of the random variables. Using the results of Propositions 1 and 2, we thus have

$$\frac{1}{C}\bar{T}^*_N(1) = \frac{M}{C}\lambda^{-1}\log(N) + \Theta(1), \tag{9}$$

and

$$\frac{M}{C}\bar{T}^*_N(M)+1 = \frac{M}{C}\lambda^{-1}[\log(N)+(M-1)\log\log(N)]+\Theta(1). \tag{10}$$

Therefore

$$\lim_{N\to\infty} \frac{\frac{M}{C}\bar{T}^*_N(M)+1}{\frac{1}{C}\bar{T}^*_N(1)} = 1,$$

and the theorem follows. ■

From Theorem 4, we observe that, as $N \to \infty$, the expected delay using the PCRR policy with $C \leq M$ channels achieves a value that is C times smaller than the optimal expected delay with a single channel. This result holds even though each node is only equipped with a single radio.

We also note that the performance of the PCRR policy can be improved by letting it skip the transmissions of packets already received by all the nodes. We can then easily prove that PCRR is optimal in the single channel and the unconstrained channel cases. However, all the results obtained for the PCRR policy in this section hold without this requirement.

c) The case $S < C$: We now show that the previous results apply also to the case $S < C$. To prove this, let the control policy consist of two stages. During stage 1, packets will be sent out in a round-robin fashion over a single channel. Once at least $(C-S)$ nodes have received all M packets, then stage 2 starts, during which PCRR is employed. Let the time the system spends in stage 1 be $T_{s1}(N)$.

We now show that $\bar{T}_{s1}(N) = \Theta(1) = o(\log(N))$. Consider a policy that selects a priori $(C-S)$ out of N nodes, then use a single channel policy to transmit the M packets to each of these nodes. The expected time for these nodes to receive all packets is $\bar{T}^*_{(C-S)}(1) = \Theta(1)$. In practice, obviously $\bar{T}_{s1}(N) \leq \bar{T}^*_{(C-S)}(1)$, thanks to the broadcast property of the wireless channel. Thus, $\bar{T}_{s1}(N) = o(\log(N))$.

d) In-order packet delivery: The results of this section, as well as that of Section V, can easily be generalized to the case where in-order packet delivery is required. For the single channel case, Proposition 1 continues to hold if the source transmits packets in order, i.e, the source starts transmitting packet $i+1$ only after all the nodes received packet i, for all $i \geq 1$. Similarly, for the unconstrained channel case, Proposition 2 holds without changes.

Now, consider the constrained channel case. Denote by $\hat{T}^*_N(C)$ the completion time using the optimal in-order packet delivery policy, for a cluster of N nodes with C channels. Clearly, $\hat{T}^*_N(C) \geq_{st} T^*_N(C)$, since the control space of in-order packet delivery is a subset of the control space of out-of-order packet delivery. Thus, Theorem 2 still holds. Theorem 3 continues to hold as well, since an in-order PCRR policy cannot perform worse (on a sample path basis) than the PCRR policy applied to the SYS'_{PCRR} system, where nodes also receive packets in order. It follows that Theorem 4 is valid for in-order packet delivery as well.

V. Analysis of Multi-Hop Cluster Chains

We next analyze the cluster chains model introduced in Section III-C. We start with the computation of a lower bound on the optimal expected delay. Then, we derive an asymptotically optimal policy, for the case where N tends to ∞. In the sequel, we will denote by $\bar{T}^*_{\mathtt{CL}}(C)$, the minimum expected time to completion of the optimal policy for a cluster chains topology with C channels available. The results presented in this section are for the case $1 \leq C \leq M$. The case $C > M$ can be treated analogously.

A. Lower Bound

Proposition 3: Consider a cluster chains network with C channels, where $1 \leq C \leq M$. Then, as $N \to \infty$,

$$\bar{T}^*_{\mathtt{CL}}(C) \geq \frac{M}{C}\lambda^{-1}\log(N) + \Theta(1),$$

where $\lambda = \log(1/p)$.

Proof: Consider the expected time to send all M packets to all the nodes belonging to cluster $(1,1)$. This quantity obviously represents a lower bound on the expected time to transmit all the packets to all the nodes in the network. Since cluster $(1,1)$ contains $\alpha_{11}N$ nodes, it will take at least $\bar{T}^*_{\alpha_{11}N}(C)$ slots to complete data dissemination to all the nodes belonging this cluster (we recall that the notation $\bar{T}^*_N(C)$ represents the optimal expected delay to disseminate M packets in a single-hop cluster of N nodes). The result then follows from a direct application of Theorem 2 and Proposition 1. ■

B. Upper Bound: The MPCRR Policy

We next propose an asymptotically optimal policy, called Multi-hop Packet-Channel Round-Robin (MPCRR). This policy proceeds in two stages. In stage 1, we ensure that at least C nodes in each cluster receive all the M packets. In stage 2, clusters are carefully grouped together so that packets can be transmitted in parallel to nodes in each group without causing collision. Thus, for each group, a subset of the nodes having received all the packets in stage 1 are selected to serve as sources. These nodes follow the PCRR transmission policy described in Section IV-B.

1) MPCRR policy stage 1: Our goal in stage 1 is to ensure that each cluster has enough nodes qualified to act as sources in stage 2. There exist many heuristics that can accomplish this process in $\Theta(1)$ time slots, on average. For example, consider a policy that selects a priori C nodes in each cluster and operates as follows. First, a source node belonging to the root cluster

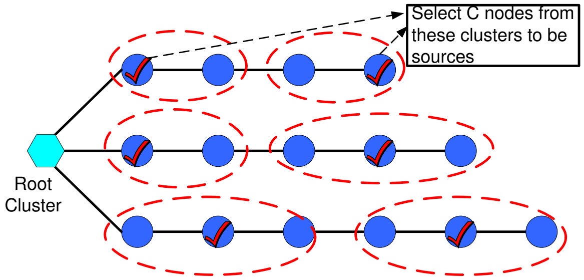


Fig. 4. Illustration of the MPCRR_S2 grouping algorithm.

transmits according to the single channel PCRR policy until all the selected nodes in each level 1 cluster receive the M packets (a level i cluster is a cluster that is i hops away from the root cluster). Then, in each level 1 cluster, one node is selected as a source and transmits according to the single channel PCRR policy to the selected nodes of its child cluster at level 2, and so forth. The expected time spent in stage 1 in this example is bounded by $W \cdot D \cdot \bar{T}^*_C(1) = \Theta(1)$ (recall that W is the total number of cluster chains and D is the length of the longest chain). In practice, this policy is not very efficient as it does not take advantage of pipelining, that is, the possibility of having multiple packets being transmitted in parallel on each chain. It is also unnecessary to select a priori which nodes will serve as sources. In the Appendix, we describe a more efficient heuristic for stage 1.

2) MPCRR policy stage 2: According to the 2-hop interference model, to avoid packet collisions, the minimum distance between two nodes transmitting simultaneously on the same channel must be three hops. Thus, the MPCRR_S2 algorithm, which implements stage 2 of the MPCRR policy, organizes clusters into groups (cf. Algorithm 2). For cluster chains of length divisible by three, MPCRR_S2 gathers every three consecutive clusters into a group. For cluster chains of length not divisible by three, MPCRR_S2 makes that sure that the groups at the two boundaries of the chain have at least two clusters and that the other groups contain three clusters. Figure 4 illustrates how MPCRR_S2 groups the clusters.

For each group, transmissions are performed by sources all belonging to the same cluster, which is referred to as a source cluster. Since source clusters are at least three hops away one from another and since every node can hear the sources of the group to which it belongs, the PCRR policy can be run simultaneously in each group. This fact guarantees that the completion time of MPCRR_S2 is smaller or equal (in a stochastic ordering sense) to the completion time of PCRR in a single cluster containing N nodes.

C. Asymptotic Optimality

Denote the expected completion time using the MPCRR policy with C channels by $\bar{T}_{MPCRR}(C)$. The following proposition characterizes the asymptotic behavior of $\bar{T}_{MPCRR}(C)$.

Proposition 4: Consider a cluster chains network with C channels, where $1 \leq C \leq M$. Then, as $N \to \infty$,

$$\bar{T}_{MPCRR}(C) \leq \frac{M}{C}\lambda^{-1}[\log(N) + (M-1)\log\log(N)] + \Theta(1),$$

where $\lambda = \log(1/p)$.

Proof: The result follows directly from the fact that stage 1 of MPCRR completes on average in $\Theta(1)$ time slots and that stage 2 completes on average at least as fast as the PCRR policy applied to a single-hop network of N nodes. ■

Algorithm 2 Multi-hop Packet Channel Round Robin Stage 2

Input:
The number of channels available, C;
Output:
Completion time, T_{MPCRR_S2};

for Each cluster chain w **do**
  Let $D(w)$ be the number of clusters in the current cluster chain, excluding the root cluster;
  Index the clusters $1, 2, .., D(w)$;
  if $D(w) \mod 3 == 1$ **or** $D(w) \mod 3 == 2$ **then**
    Group the first two clusters together, and let cluster 1 be a *source cluster*;
  else
    Group the first three clusters together, and let cluster 2 be a *source cluster*;
  end if
  Continue to group the rest of the clusters into groups of three clusters, except for the last group which may have two or three clusters. For each group, the cluster located three hops away from the *source cluster* of the previous group is designated as the *source cluster*;
  In each *source cluster*, select C nodes possessing all the packets to act as sources.
end for
Implement the PCRR(M, C) policy in each group using the selected source nodes until all the nodes have received all the packets.

The following theorem states the asymptotic optimality of the MPCRR policy. Its proof directly follows from Propositions 3 and 4.

Theorem 5: Consider a cluster chains network with C channels with $1 \leq C \leq M$. Then

$$\lim_{N\to\infty} \frac{\bar{T}_{MPCRR}(C)}{\bar{T}^*_{\mathrm{C1}}(C)} = 1.$$

VI. Numerical Results

We next present numerical results to evaluate the performance of the PCRR policy in networks of finite size. Our simulation experiments focus on the following aspects: *(i)* the gain achieved by PCRR when exploiting multiple channel resources in single cluster and cluster chains topologies; *(ii)* the performance of PCRR compared to that of the optimal policy; and *(iii)* the performance of PCRR under heterogeneous packet loss. The results represent an average over 1000 identical simulations.

A. Multi-channel Gain

1) Single Clusters: We consider the case of disseminating a file consisting of $M = 20$ packets, when employing the PCRR transmission policy on a single cluster topology with packet loss probability $p = 0.3$.

In the first set of simulations, we assume $S \geq C$. Figure 5 depicts the average completion time of PCRR as a function of the number of nodes in the network, for the cases $C = 1, 2, 5, 10$. As expected, in each case, the average completion time increases logarithmically with N (note that the x-axis in the figure is on a logarithmic scale). The figure shows that significant reduction in the average completion time can be achieved with only two channels. Through a linear regression, we observe that the slope of the curves corresponding to $C > 1$ channels are about C times smaller than that corresponding to a single channel, as predicted by the asymptotic analysis.

Next, we consider the scenario where there is initially only one source that possesses all the packets. In this case, PCRR is applied using C' channels, where $C' = \min(S, C, M)$ is a variable that is updated in real time. Whenever a nodes finishes receiving all the M packets, the value of S increases and so does that of C' until S equals C or M.

Figure 6 illustrates this scenario. As expected from our analysis, the performance of PCRR in this case is similar to the case where $S \geq C$. Interestingly, when $C > 1$, we find that the expected completion time does not always increase monotonically with the number of nodes. This is because on the one hand, it takes more time to disseminate the file to a larger number of nodes, but on the other hand, with more nodes present, it takes less time to find nodes that have received all the packets and can serve as sources.

In Figure 7, we evaluate the effect of the packet loss probability p on the completion time of the PCRR policy, for different number of channels. Specifically we set $N = 1000$, $M = 40$, $C = 1, 2, 5, 10$, and $S \geq C$ while p varies from 0.001 to 0.5. The simulation results indicate that the absolute gain achieved using multiple channels becomes more significant as p increases. This result is expected since the completion time always takes exactly M time slots when $p = 0$.

2) Cluster Chain: Figure 8 evaluates the performance of the MPCRR policy on a cluster chain topology with parameters $D = 10$, $W = 1$ and $\alpha_{wd} = \frac{1}{WD}$. That is, the network is a composed of a single 10-hop cluster chain, where each cluster contains the same number of nodes. We assume $S \geq C$, $M = 20$, and $p = 0.3$. Stage 1 of the MPCRR policy is implemented using the MPCRR_S1 algorithm described in the Appendix.

The figure shows that MPCRR fully exploits the multiple channels resources of the nodes, with a reduction in the average delay roughly proportional to the number of channels available. Comparing Figure 5 with Figure 8, we notice that although these figures correspond to different network topologies, the expected completion times are close for networks of same size. This is due to the fact that stage 2 of the MPCRR policy dominates the average completion time in a cluster chain network, which we have shown to be asymptotically

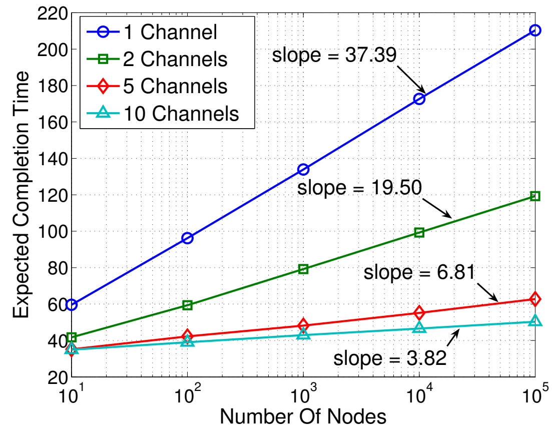


Fig. 5. Performance of the PCRR algorithm with different number of channels: $M = 20$, $p = 0.3$, $S \geq C$.

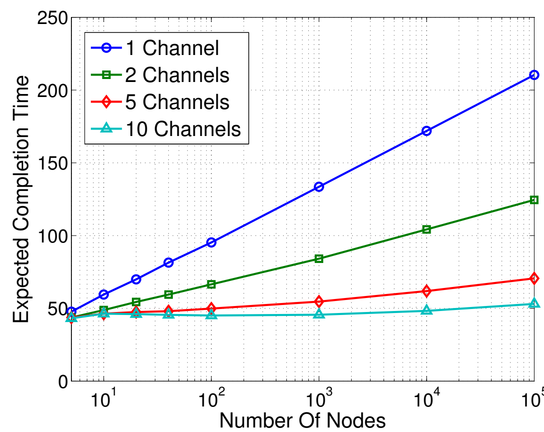


Fig. 6. Performance of the PCRR algorithm with different number of channels: $M = 20$, $p = 0.3$, $S = 1$.

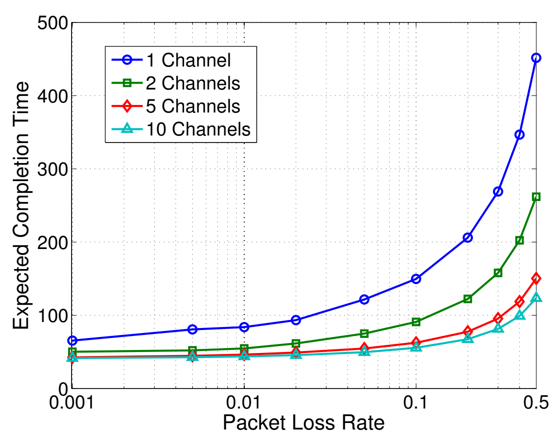


Fig. 7. Multi-channel gain under different packet loss rates: $M = 40$, $N = 1000$, $S \geq C$.

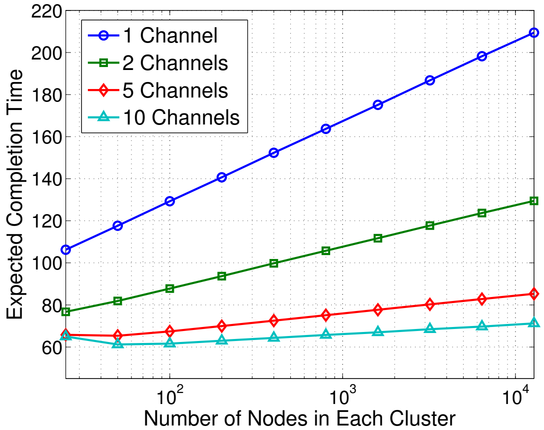


Fig. 8. Performance of the MPCRR algorithm with different number of channels in a multi-hop cluster chain network: $M = 20$, $D = 10$, $W = 1$, $p = 0.3$, $S \geq C$.

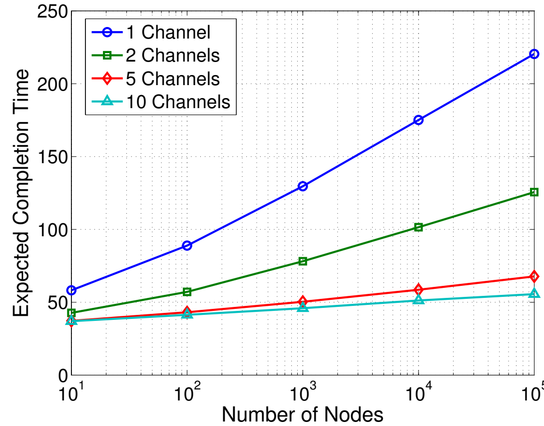


Fig. 10. Multi-channel gain of PCRR under heterogeneous packet loss rates: Single cluster, $M = 20$, $S \geq C$.

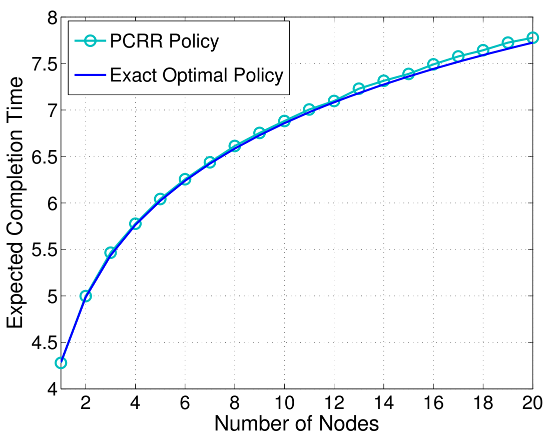


Fig. 9. Comparison of PCRR and the optimal policy for a single cluster with in-order packet delivery: $M = 3$, $C = 2$, $p = 0.3$, $S = 2$.

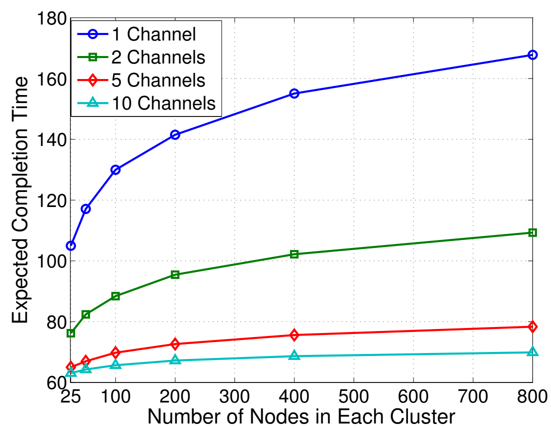


Fig. 11. Performance of the MPCRR algorithm with different number of channels in a multi-hop cluster chain network under heterogeneous packet loss rates: $M = 20$, $D = 10$, $W = 5$, $S \geq C$, and $\alpha_{wd} = \frac{1}{WD}$.

equivalent to the average completion time in a single cluster network of the same size.

B. Comparison with the Optimal Policy

We next compare the performance of the PCRR policy with the optimal policy obtained by solving Eq. (1). As discussed in section III-B, solving the optimal policy is computationally involved. We thus consider a small example where $M = 3, C = 2, p = 0.3, N = 1..20$. Even then, the state and control spaces in the dynamic programming problem are huge. Therefore, we consider the case for which in-order packet delivery is required, which substantially reduces the size of the state and control spaces. Figure 9 shows that the performance of PCRR is close to that of the optimal policy. Although the PCRR policy is proven to be asymptotically optimal only when N is very large, it performs very well compared to the exact optimal policy, even in small networks.

C. Heterogeneous Packet Loss Rates

Our analytical results apply to the case where the packet loss probability is fixed. Yet, in practice, the rate of packet losses may differ from node to node due to various factors (e.g. channel fading, antenna sensitivity, etc). It is therefore of interest to evaluate the behavior of PCRR and MPCRR policies under heterogeneous packet loss. We therefore run simulations for the case where a different packet loss probability, chosen uniformly at random between 0 and 0.4, is associated to each node and kept fixed throughout the simulation.

Simulation results for a single cluster topology are shown in Figure 10. The results resemble those obtained under homogeneous packet loss. The PCRR policy efficiently exploits multi-channel resources and achieves a significant reduction in completion time. Similar results are shown in Figure 11 for MPCRR and a cluster chains topology. These results could be inferred from Theorem 2 and Theorem 3, in which the sample path arguments used for the proof apply to heterogeneous packet loss probabilities as well.

VII. Practical Implementation of the PCRR Policy

The main emphasis of this paper is on characterizing the theoretical performance limits of data dissemination in sensor networks. As such, practical considerations, such as control overhead and channel switching latency, are not explicitly

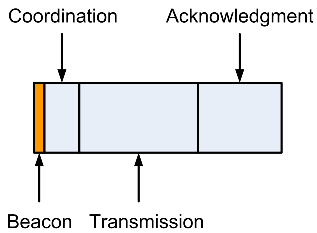


Fig. 12. Implementation the PCRR policy: time allocation in each frame.

captured in our analytical model. The goal of this section is to show that the PCRR policy is nevertheless amenable to practical implementation and able to yield significant performance gains with respect to a single channel baseline protocol. We outline a protocol implementation of the PCRR policy for the case of a single-hop network and present simulation results obtained with the TOSSIM simulator, a bit-level network simulator designed specifically for TinyOS-based sensor networks [14].

A. Protocol Design

Our protocol divides network nodes into three classes: *(i)* a base node that periodically broadcasts coordination information to all the nodes; *(ii)* source nodes that disseminate packets; *(iii)* destination nodes that are packet recipients. Note that a destination node may become a source node once it receives all the packets.

The time axis is divided into frames of variable length. Each frame consists of four stages, beacon, coordination, transmission, and acknowledgment, as shown in Figure 12. At the beginning of each frame, all the nodes listen to the same, default channel. The base node sends a short beacon message (preamble) to synchronize all the nodes. Next, the base node sends a coordination message. This message contains the list of source nodes and channels on which they should operate. It also includes the number of time slots the PCRR policy will be run on during the transmission stage (we explain below how this number is determined). During the transmission stage, the source nodes transmit packets on the channels assigned to them. The destination nodes use the coordination information sent by the base node to determine which packet will be sent on which channel, according to the PCRR policy. The protocol imposes a short latency between each transmission to allow destination nodes to switch channels. Finally, in the acknowledgment stage, all the nodes switch back to the default channel. If a node is missing one or more packets, it triggers a timer with a random expiration time. Once the timer expires, the node broadcast a NACK message, which is a request vector specifying which packets are needed.

To reduce the number of NACKs in the acknowledgment stage, we borrow the suppression mechanism approach from [2]. When overhearing a NACK request vector, a node takes its union with all the other request vectors overheard so far. Once the timer expires, the node compares the union of the previous request vectors to its own request vector. If this vector is a subset of the union, then the node suppresses its NACK. Otherwise it broadcasts the NACK message as scheduled. In order to take full advantage of this mechanism, nodes requesting a large number packets are likely to be assigned a shorter expiration time than those needing a small number of packets. This is achieved by setting the timer value to $t_{timer} = \left(\tau \frac{M_r}{M} + \text{rand}(\omega)\right) t_{ack}$, where M_r is the number of packets that a node has received up to the current frame, rand(ω) is a random number uniformly distributed between 0 and ω, and t_{ack} is the time length of the acknowledgment stage.

The number of time slots that the PCRR policy should be run on during the transmission stage of frame t corresponds to the maximum number of packets (say M_{t-1}) requested by a node (say node n) during the acknowledgment phase of frame $t-1$. This approach helps in reducing the overhead, since it must take at least M_{t-1} slots for node n to receive all of its missing packets. The number of time slots in the first frame of the transmission stage should be set to M.

If a node finishes receiving all the packets and finds out that some channels are not utilized because of a lack of source nodes, then it sends a notification to the base node requesting to become a source node. This notification message is sent in a similar way to that of a NACK request vector. The notification message is suppressed if a node finds out that the number of source nodes is already large enough, i.e., $S \geq C$.

We briefly outline how the above protocol can be extended to cluster chains. First, C nodes are selected in each cluster to serve as potential relaying sources. The M-packet file is disseminated to these nodes, using a procedure similar to those explained in Section V-B and in the Appendix. Once all the potential sources in a cluster finish receiving all the packets that they need, they use MPCRR_S2 (see Algorithm 2) to determine if they should transmit. If yes, they implement the same PCRR protocol as described above for single-hop networks.

B. TOSSIM Evaluation

The multi-channel PCRR protocol described in the previous section involves two types of overhead absent in a single channel protocol, namely the coordination stage at the beginning of each frame and the switching latency between channels. The goal of this section is to evaluate whether the benefit of exploiting multi-channel resources using the PCRR policy prevails over the overhead associated with it.

The evaluation is performed using the TOSSIM simulator based on the following parameters. The size of the file to be disseminated is 10KB. This file is initially held only by the base station. The file is divided into $M = 20$ packets of length 516 bytes each (512 bytes of data + 4 bytes of overhead). The size of NACK packets is 16 bytes, including overhead. In each frame, the beacon stage lasts for 0.1 sec, the coordination stage lasts for 0.5 sec, and the acknowledgment stage lasts for 2.0 sec. The channel switching latency is set to 50 msec, based on our own measurements on MICA2 motes. Hence, at each slot of the transmission stage, a source node must wait this amount of time before starting to transmit. Similarly, the acknowledgment stage starts only after a waiting period of 50 msec. The parameters of the random timer are $\tau = 0.5$

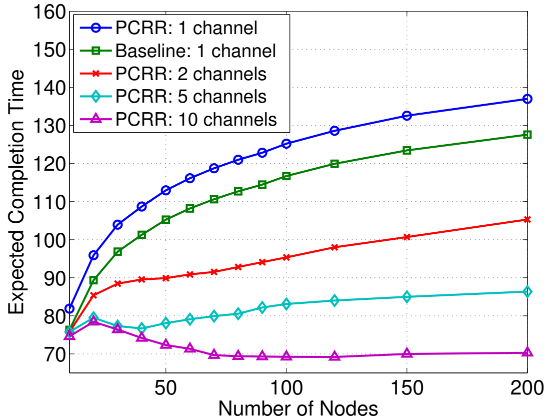


Fig. 13. Performance of the PCRR policy with the presence of control overhead: $M = 20$, $p = 0.3$, $S = 1$.

and $\omega = 0.25$. For the PCRR policy, simulation experiments are run for the case of $C = 1, 2, 5, 10$ channels. We also evaluate the performance of a baseline single channel protocol that is similar to a single channel PCRR protocol, but without coordination stage and channel switching latencies overhead.

The results of the experiments, depicting the average completion time as a function of the network size, are presented in Figure 13. Each point in the figure represents an average taken over 1000 identical experiments. As expected, for the case $C = 1$, PCRR performs worse than the baseline protocol. However, as soon as $C \geq 2$, the PCRR protocol significantly outperforms it. For instance, for the case of $N = 100$ nodes, and $C = 2, 5$ and 10 channels, PCRR achieves a savings of about 20%, 30%, and 40%, respectively, with respect to the baseline protocol. The non-monotonic nature of the completion time is due to the fact that the system starts with a single source, as explained previously for Figure 6. In summary, the results of our evaluation show that the PCRR policy is not only asymptotic optimal under theoretical settings but also greatly improves the performance and scalability of data dissemination in practice.

VIII. Concluding Remarks

In this paper, we formalized the problem of data dissemination in multi-channel single radio sensor networks with random packet loss. We showed that, for an arbitrary topology, the problem of minimizing the expected delay of data dissemination can be cast as a stochastic shortest path problem. Interestingly, due to its special structure, this problem can be related to a deterministic shortest path problem and solved using any label-based algorithms, such as Dijkstra. However, the computational burden necessary to derive the optimal policy is generally too high to be practical.

Therefore, to obtain a more tractable and insightful solution, we restricted our attention on two important classes of topologies, namely single hop clusters and multi-hop cluster chains, and conducted a large network asymptotic analysis for these topologies. Based on extreme-value theory and coupling arguments, we proved the asymptotic optimality of a simple policy, called Packet Channel Round Robin (PCRR), and its multi-hop generalization, called Multi-hop Packet Channel Round Robin (MPCRR). With C channels available, we showed that these policies are capable of reducing the expected delay by a factor of C, as if each node were equipped with C radios. Numerical simulations confirmed the applicability of these findings to small and moderately-sized networks.

The PCRR policy has the desirable property of not depending on the network state (i.e., the specific set of packets received by each node), contrarily to the optimal policy. To demonstrate its practical use and benefit, we presented an implementation and evaluation of this policy for a single-hop cluster. Using the TOSSIM simulator, we showed that the overhead associated with the implementation is limited and scales with the network size. The simulations indicated that the PCRR implementation significantly reduces the average delay with respect to a single channel baseline protocol, e.g., by a margin of 30% when using 5 channels in a 100-node network.

In summary, this paper shows that the multichannel transceiving capability of single radio sensor nodes can be exploited to achieve significant reduction in the delay of data dissemination. The findings are expected to be useful to other types of multi-channel single radio wireless networking technologies, such as IEEE 802.11 wireless LANs. The paper leaves many interesting problems for future work. This includes extending the analysis to the case where clusters belonging to different chains interfere as well as devising asymptotically optimal policies for general topologies.

Acknowledgments

The authors would like to thank Dr. Xiangping Qin and Dr. Ari Trachtenberg for fruitful discussions and for their contributions to a preliminary version of this work [1].

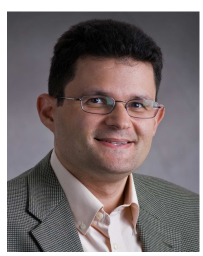
Appendix

We describe here an efficient heuristic, denoted MPCRR_S1, to implement stage 1 of the MPCRR policy. The basic rules of MPCRR_S1 are the following: *(i)* packets are transmitted in order; *(ii)* a parent cluster is tentatively scheduled to send out a packet (say packet m) to its child cluster(s), only if at least C nodes in the parent cluster already possess packet $1, 2, \ldots, m$, and fewer than C nodes possess these packets in one or more of the child clusters; *(iii)* among the tentative transmissions scheduled by rule *(ii)*, if there is a channel contention, a descendant cluster is given priority over an ancestor clusters if it scheduled to transmit a packet with a lower index; otherwise the ancestor cluster is given priority over the descendant cluster; *(iv)* When $C > 1$, clusters at level $4n, 4n+1$ transmit on channel 1 and clusters at level $4n+2, 4n+3$ transmit on channel 2, where $n = 0, 1, 2, \ldots$. This rule allows simultaneous transmissions by nodes that are two-hops apart.

As $N \to \infty$, it is straightforward to show that, with probability one, it takes only one time slot for a packet to be transmitted from all level l clusters to all level $(l+1)$ clusters. For the case $C = 1$, one can then show that the MPCRR_S1 policy is identical to the optimal, deterministic scheduling policy for a lossless chain network with omnidirectional antennas [24]. This policy completes in $3(M-1)+D$ slots. Similarly, for the case $C>1$, MPCRR_S1 policy is identical to the optimal, deterministic scheduling policy for a lossless chain network with directional antennas, which completes in $2(M-1)+D$ time slots.

REFERENCES

[1] D. Starobinski, W. Xiao, X. Qin, and A. Trachtenberg, "Near-optimal data dissemination policies for multi-channel, single radio wireless sensor networks," in *IEEE INFOCOM '07*, Anchorage, AK, May 2007.
[2] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *SenSys '04*, 2004.
[3] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MobiCom '00*. Boston, MA, USA: ACM, 2000, pp. 56–67.
[4] S. S. Kulkarni and L. Wang, "MNP: Multihop network reprogramming service for sensor networks," in *ICDCS '05*, 2005.
[5] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Wireless Networks*, vol. 8, no. 2/3. Kluwer Academic Publishers, 2002, pp. 153–167.
[6] W. Xiao and D. Starobinski, "Poster abstract: Exploiting multi-channel diversity to speed up over-the-air programming of wireless sensor networks," in *SenSys '05*, San Diego, CA, USA, 2005, pp. 292–293.
[7] R. Simon, L. Huang, E. Farrugia, and S. Setia, "Using multiple communication channels for efficient data dissemination in wireless sensor networks," in *MASS '05*, 2005.
[8] L. Wang and S. S. Kulkarni, "*Gappa*: Gossip based multi-channel reprogramming for sensor networks." in *DCOSS*, San Francisco, CA, USA, June 18-20, 2006, pp. 119–134.
[9] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher, "MMSN: Multi-frequency media access control for wireless sensor networks," in *INFOCOM '06*, Barcelona, Spain, April, 2006.
[10] M. J. Miller and N. H. Vaidya, "Leveraging channel diversity for key establishment in wireless sensor networks," in *INFOCOM '06*, Barcelona, Spain, April 2006.
[11] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.
[12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS '00*, Washington, DC, USA, 2000, p. 8020.
[13] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power efficient gathering in sensor information systems," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3, 2002, pp. 1125–1130.
[14] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *SenSys '03*. Los Angeles, CA, USA: ACM Press, 2003, pp. 126–137.
[15] A. Nasipuri, J. Zhuang, and S.R.Das, "A multichannel CSMA MAC protocol for multihop wireless networks," in *WCNC '99*, 1999.
[16] A. Tzamaloukas and J. Garcia-Luna-Aceves, "Channel-hopping multiple access," *ICC '00*, vol. 1, pp. 415–419, June 2000.
[17] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *MobiCom '04*, Philadelphia, PA, USA, 2004, pp. 216–230.
[18] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *MobiHoc '04*. Roppongi Hills, Tokyo, Japan: ACM, 2004, pp. 222–233.
[19] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *INFOCOM '05*, vol. 3, 2005, pp. 2223 – 2234.
[20] R. Iyengar, V. Sharma, K. Kar, and B. Sikdar, "Analysis of contention-based multi-channel wireless MAC for point-to-multipoint networks," in *WOWMOM '06*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 453–455.
[21] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: Impact of number of channels and interfaces," in *MobiCom '05*. Cologne, Germany: ACM Press, 2005, pp. 43–57.
[22] C.-J. Su and L. Tassiulas, "Broadcast scheduling for information distribution," in *INFOCOM '97*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 109–117.
[23] N. H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," in *Wireless Networks*, vol. 5, no. 3. Hingham, MA, USA: Kluwer Academic Publishers, 1999, pp. 171–182.
[24] C. Florens and R. McEliece, "Scheduling algorithms for wireless ad-hoc sensor networks," in *GLOBECOM '02*, vol. 1, Taipei, Taiwan, 17-21 Nov. 2002, pp. 6–10.
[25] S. K. Kasera, G. Hjálmtýsson, D. F. Towsley, and J. F. Kurose, "Scalable reliable multicast using multiple multicast channels," *IEEE/ACM Transactions on Networking (TON)*, vol. 8, no. 3, pp. 294–310, 2000.
[26] E. Hahne, "Round-robin scheduling for max-min fairness in data networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 7, pp. 1024–1039, 1991.
[27] K. Bogineni, K. M. Sivalingam, and P. W. Dowd, "Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 4, pp. 590–604, 1993.
[28] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Transactions on Networking (TON)*, vol. 4, no. 3, pp. 375–385, 1996.
[29] S. Ramabhadran and J. Pasquale, "Stratified round robin: a low complexity packet scheduler with bandwidth fairness and bounded delay," in *SIGCOMM '03*. Karlsruhe, Germany: ACM, 2003, pp. 239–250.
[30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.
[31] D. P. Bertsekas and J. N. Tsitsiklis, "An analysis of stochastic shortest path problems," *Mathematics of Operations Research*, vol. 16, no. 3, pp. 580–595, 1991.
[32] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 366–379, Oct.-Dec. 2004.
[33] R. Krishnan and D. Starobinski, "Efficient Clustering Algorithms for Self-Organizing Wireless Sensor Networks," *Journal of Ad-Hoc Networks*, pp. 36–59, January 2006.
[34] J. Misic and V. Misic, "Building multicluster 802.15.4 networks: issues and open problems," *Wireless Communications, IEEE*, vol. 15, no. 4, pp. 74–78, Aug. 2008.
[35] S. M. Ross, *Stochastic Processes*. Wiley, January 1995.
[36] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja, *A First Course in Order Statistics*. John Wiley & Sons, Inc, 1992.
[37] R. M. Young, "Euler's constant," *Math. Gazette*, vol. 75, pp. 187–190, 1991.
[38] S. Kang and R. F. Serfozo, "Extreme values of phase-type and mixed random variables with parallel-processing examples," *Journal of Applied Probability*, vol. 36, pp. 194–210, Nov. 1999.
[39] J. Pickands III, "Moment convergence of sample extremes," *The Annals of Mathematical Statistics*, vol. 39, no. 3, pp. 881–889, June 1968.



David Starobinski received his Ph.D. in Electrical Engineering (1999) from the Technion-Israel Institute of Technology. In 1999-2000 he was a visiting post-doctoral researcher in the EECS department at UC Berkeley. In 2007-2008, he was an invited Professor at EPFL (Switzerland). Since September 2000, he has been with Boston University, where he is now an Associate Professor.

Dr. Starobinski received a CAREER award from the U.S. National Science Foundation and an Early Career Principal Investigator (ECPI) award from the U.S. Department of Energy. His research interests are in the modeling and performance evaluation of high-speed, wireless, and sensor networks.

Weiyao Xiao received B.E. degree from Harbin Institute of Technology, Harbin, China and M.S. degree from Boston University in 2004 and 2007 respectively. Currently, he is working towards his Ph.D. degree in Electrical and Computer Engineering, also at Boston University. His research interest centers around wireless and sensor networks.