# Effect of Topology Knowledge on Opportunistic Forwarding

Prithwish Basu
BBN Technologies, Cambridge MA
Email: pbasu@bbn.com

Saikat Guha
BBN Technologies, Cambridge MA
Email: sguha@bbn.com

***Abstract*—Opportunistic forwarding is a simple scheme for packet routing in delay tolerant networks such as duty cycling sensor networks in which reducing energy consumption is a principal goal. While it is simple and can be analytically characterized, it suffers from a high end-to-end latency. In this paper we show how this latency can be drastically reduced if nodes have limited knowledge of network topology (that can be achieved by scoped dissemination of link state information), and hence deriving a hybrid routing protocol in such networks. We give an analytical formulation of end-to-end latency between any pair of nodes in such duty cycling networks as the scope of topology dissemination is varied. We borrow from our prior results derived from spectral graph theory to derive exact expressions for mean latency as a function of various network and protocol parameters such as size, duty cycle probability, and scope of link state dissemination. We show that these analytical expressions agree very well with simulation results. We also show how this latency analysis can be coupled with overhead analysis to determine good values of topology dissemination scope.**

## I. Introduction

Routing in wireless multihop networks has been a topic of intense research for the past several decades. A cornucopia of routing protocols has emerged over the past two decades. The space of protocols can be classified into many categories such as reactive (DSR, AODV), proactive (OLSR, HSLS), hybrid (ZRP, SHARP), those that require geo-location information for efficiency (GPSR and variants) to list only a small fraction of them. The space of requirements in multihop networking is so diverse that no single protocol works better than all the rest in all different scenarios, with the scenario space being defined by attributes such as network size (small vs. large or dense vs. sparse), link dynamics (deterministic vs. stochastic), mobility patterns (regular vs. irregular/random), traffic patterns (local vs. global), traffic load (low vs. high), optimization metrics (shortest path vs. lowest latency vs. lowest energy consumption) etc. Most researchers use simulation as the primary means of conducting performance evaluation of these protocols. While the complexity of protocols amply justifies realistic simulation and real implementation and deployment of protocols, this has resulted in a general lack of rigor in the literature in the analysis of the fundamental properties of these protocols.

Since analytical frameworks, albeit simplistic, can give deeper insights into the fundamental properties of protocols than simulations or real implementations, we propose some analytical techniques to study a simple class of routing protocols in this paper and show how effective such techniques can be to study a fundamental property of routing protocols – path length or latency. The specific class of protocols that we study in this paper is that of opportunistic forwarding in duty cycling networks. Duty cycling (or sleep-scheduling) in energy-constrained wireless ad hoc networks refers to the process of turning radio transceivers OFF for conserving battery energy that is wasted by idle listening, and then turning them back ON when there is an opportunity to participate in network communication. Several mechanisms have been proposed for performing duty cycling in the literature [9], [6], [7]. In the *pseudo-random duty cycling* model [7], by exchanging a small set of parameter values (such as seed and cycle position of a pseudo-random-number-generator, and wake-up probability $p$), each node can construct the exact ON/OFF state information of its neighbors, thus allowing for energy-efficient communication.

In such energy-constrained networks with time-varying topology caused by duty cycling, it is often difficult to execute traditional MANET routing protocols because disseminating/gathering topology state from the entire network is a time and energy consuming expenditure. Moreover, at low duty cycling probabilities, dissemination of topology information (a step used by the link-state family of schemes) or that of route queries and responses (steps used by reactive protocols) can take a significant amount of time which can render the topology information less useful by the time it is gathered. Hence stateless opportunistic forwarding (e.g. forward the packet to the first neighbor that would wake up, or *first-contact* routing in DTN scenarios) is a simple technique which can be used to deliver packets under such resource constrained scenarios [2]. However, a significant drawback of this scheme is the high end-to-end latency suffered due to the random walk component.

Clearly, a natural mechanism for improving the latency performance of this routing scheme is to utilize some topology

[1]This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

information for routing. In this paper, we study two such scenarios: (a) what is the effect of routing topology knowledge on the latency of opportunistic forwarding in two-dimensional grid lattices[2]; and (b) what is the effect of partial knowledge of routing topology on latency of opportunistic forwarding in general network topologies, i.e. when nodes limit their link state topology dissemination to $k$ hops?

Our contributions in this paper are the following:

1) We give exact expressions for expected latency of random walks as a function of the size of the grid $n$, and the duty cycling probability $p$.
2) We derive an analytical expression for expected latency of random walks in the regular line lattice with $k$-hop topology information.
3) We derive a general expression for expected latency of random walks for any general network topology with $k$-hop topology information. This could help a deployer of routing protocol choose a value of $k$ that optimizes the overhead vs. latency tradeoff.
4) The analytical results mentioned above are shown to agree very well with simulations. To the best of our knowledge this is the first paper that analytically studies the dependence of scope $k$ on the performance of random walks in wireless networks, with or without duty cycling.
5) We determine good values of $k$ by performing joint analysis of latency and overhead.

### A. Related Work

Lu et al. have studied the problem of finding minimum end-to-end delay paths through duty cycling networks [6]. They do not attempt to give expressions for latency, nor do they study how limited topology knowledge would affect performance.

Our prior work was focused on deriving exact expressions for latency of random walks between any source-destination pair in a finite graph. We studied the duty cycling case in particular [2] and then generalized it to general weighted graphs with node-specific sojourn times [3]. However we only studied purely stateless forwarding in the aforementioned papers and did not consider the effect of topology knowledge.

The Hazy-Sighted Link State routing (HSLS) protocol which belongs to a family of "fuzzy-sighted" approaches [8] adapts the scope of link-state broadcasts over space and time to significantly reduce the overhead of flat link-state routing. They give scaling laws for overhead in terms of network size but do not perform exact analysis of routing on any given topologies (which is the focus of our paper).

Weak State Routing (WSR) is a recently proposed routing mechanism for large-scale highly dynamic networks [1]. WSR uses random directional walks biased occasionally by weak indirection state information (which is interpreted not as absolute truth, but as probabilistic hints) in intermediate nodes. Nodes only have partial information about the region a destination node is likely to be. While the scaling laws for overhead of WSR are interesting, it uses geographic state information rather than pure link state information which makes it unsuitable for general multihop networks – the ones we are studying in this paper. Moreover the WSR paper does not analyze the latency of the route for a given network topology.

[2]The routing algorithm here is Manhattan routing.

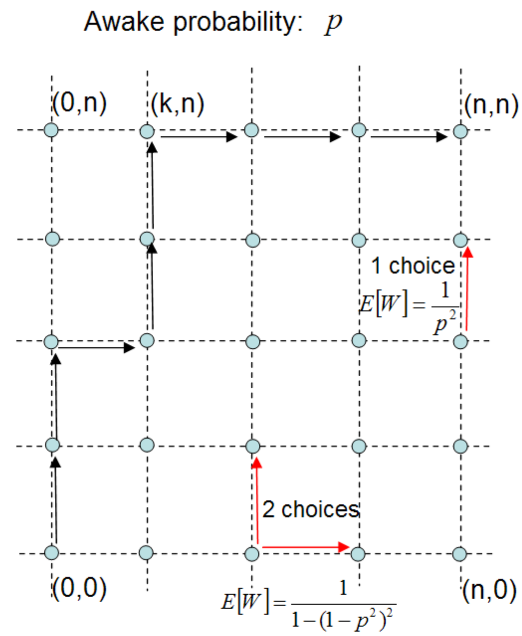


Fig. 1. Latency in a two-dimensional lattice with Manhattan routing

## II. Effect of Topology Knowledge on Random Walks: Manhattan Routing Scenario

We first study how random walk performs in the presence of the knowledge of routing. In particular we study the Manhattan grid scenario where a packet can only be forwarded to an awake neighbor located toward north or east.

Consider a lattice of $(n+1)\times(n+1)$ nodes labeled from $(0,0)$ to $(n,n)$. The length of all *shortest Manhattan* paths from the south-east corner to the north-west corner is $2n$. However, different paths could have different latencies since there may be a different number of neighbors that are available for forwarding at specific locations in each of the paths. For example, Figure 1 illustrates that once the path hits either the easternmost column or the northernmost row, each segment only allows one choice for forwarding, whereas there are 2 choices for forwarding at all other locations. We compute the expected latency incurred along a path of length $2n$ on this 2D lattice (assuming that all paths are composed of *north* and *east* segments alone).

In segments where only one choice is available for forwarding, the expected waiting latency in slots is given by $E[W]=\frac{1}{p}$ (since the latency random variable obeys the Geometric distribution). In segments where 2 choices are available for forwarding, the computation of the expected latency is a little more complicated. If the waiting time for each of the two choices is modeled as two i.i.d. discrete random variables $X$ and $Y$, then the resultant random variable for waiting time is $Z = \min(X, Y)$. Since $X$ and $Y$ are geometrically distributed with parameter $p$, then $Z$ is also geometrically distributed with parameter $1-(1-p)^2$. Hence the mean waiting time in the segments of 2 choices is given by $E[W] = \frac{1}{1-(1-p)^2}$.

Now we compute how many segments in each path of length $2n$ are of each of the above two categories. A well known result indicates that the number of paths on a 2D lattice from point $(0,0)$ to $(a,b)$ is given by $\binom{a+b}{b}$. Hence the number of paths from $(0,0)$ to $(n,n)$ is $\binom{2n}{n}$ which is $(n+1)C_n$ where $C_n$ is the $n^{th}$ Catalan number. Now half of these paths enter the northernmost row from the south, i.e, at some $(k,n), k \in [0, n-1]$, and the other half enter the easternmost row from the west, i.e., at some $(n,k), k \in [0, n-1]$. A path that enters $(k,n)$ from $(k, n-1)$ has $n-k$ segments with one choice and $n+k$ segments of 2 choices, and there are $\binom{n+k-1}{k}$ such paths. This is because this quantity is equal to the number of paths from $(0,0)$ to $(k, n-1)$. Similarly there are $\binom{n+k-1}{k}$ paths that enter $(n,k)$ from $(n-1,k)$.

Now we observe that all Manhattan paths from $(0,0)$ to $(n,n)$ are not equiprobable. This is intuitively clear since a path that follows the edge of a grid has an equal number of 2-choice vertices and 1-choice vertices; on the other hand, a path that mostly passes through the core of the grid has many more 2-choice vertices than 1-choice vertices. Once a path hits a 1-choice vertex, its fate is sealed and it has no more choices until it reaches the destination. As mentioned earlier, a path that enters $(k,n)$ from $(k,n-1)$ has $n-k$ 1-choice vertices and $n+k$ 2-choice vertices; therefore the probability of being on this particular path is given by $p_k = \frac{1}{2^{n+k}}$.

If we consider paths hitting both the north boundary and the east boundary of the grid, the number of such paths is given by $m_{n,k} = 2\binom{n+k-1}{k}$. We can easily verify that the probabilities all add up to 1 since the following holds:

$$\sum_{k=0}^{n-1} p_k m_{n,k} = \sum_{k=0}^{n-1} \frac{1}{2^{n+k-1}} \binom{n+k-1}{k} = 1$$

Therefore, the mean latency over all paths of length $2n$ is given by:

$$E[L(P_{2n})] = \sum_{k=0}^{n-1} \frac{1}{2^{n+k-1}} \binom{n+k-1}{k} \times \left\{ \frac{n+k}{1-(1-p)^2} + \frac{n-k}{p} \right\} \quad (1)$$

We use standard combinatorial identities to derive a closed form expression for Equation 1:

$$\sum_{k=0}^{n-1} \frac{n+k}{2^{n+k-1}} \binom{n+k-1}{k} = 2n - \frac{n}{2^{2n-1}} \binom{2n}{n} \quad (2)$$

$$\sum_{k=0}^{n-1} \frac{n-k}{2^{n+k-1}} \binom{n+k-1}{k} = \frac{n}{2^{2n-1}} \binom{2n}{n} \quad (3)$$

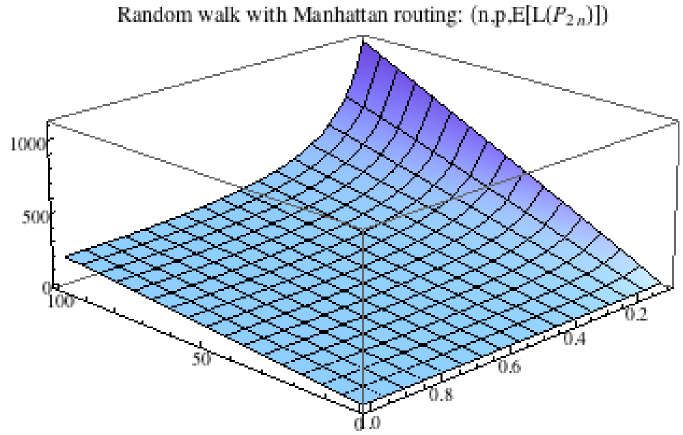


Fig. 2. Latency in a two-dimensional lattice ($n$ is half of the Manhattan distance between source and destination nodes)

Therefore, substitution of Equations 2 and 3 into Equation 1 yields:

$$E[L(P_{2n})] = \frac{1}{1-(1-p)^2} \left\{ 2n - \frac{n}{2^{2n-1}} \binom{2n}{n} \right\} + \frac{1}{p} \frac{n}{2^{2n-1}} \binom{2n}{n} \quad (4)$$

A quick sanity check indicates that as $p \to 0$, $E[L(P_{2n})] \to \infty$ and for $p = 1$, $E[L(P_{2n})] = 2n$ which is expected.

Figure 2 plots the latency expression in Equation 4 as a function of $n$ and $p$. The linear behavior in $n$ can be observed for a fixed value of $p$. However, for small values of $p$, the latency has a $\sim \frac{1}{p}$ dependence.

Incorporating independent link failures in the model is easy, e.g., if the link reliability is $\gamma$, and is independent of the node duty cycles and reliability of other links or interference caused by traffic, then it suffices to replace $p$ by $p\gamma$ in Equation 4.

## III. Effect of Partial Topology Knowledge on Random Walks

In this section we investigate the scenario where only a limited amount of topology information is available around the destination node. There are several reasons why this can happen:

1) Smart link state protocols such as Hazy Sighted Link State (HSLS) [8] disseminate the link state information more frequent to nearby nodes and less frequently to farther away nodes. HSLS performs this trick in order to reduce the control overhead.
2) A destination node may decide to broadcast link state information (i.e. state of the links between itself and its neighboring nodes) only to nodes within a limited scope. It may do that for scalability reasons since frequent network wide broadcasts can cause severe congestion in the network due to broadcast storms (even in the HSLS scenario).

In the first case, a node located topologically far away from the destination typically does not have any routing information about the destination node, hence it needs to find the latter using mechanisms other than link state routing. In networks such as duty cycling sensor networks or disruption prone networks, opportunistic forwarding algorithms are useful since they require little or no state to make forwarding decisions. For example, a node can give the packet to the first neighbor that is supposed to wake up or the first neighbor with whom a link is established or reestablished. Latency of such algorithms have been studied before [2], [3] using techniques from spectral graph theory.

If each node X disseminates its own link state information to nodes within $k$ hops, then nodes in the $k$-hop neighborhood of X are able to route a packet to X along a minimum hop route of length $k$ using this information.

Since nodes outside the $k$-hop neighborhood of the destination may need to send a packet to the latter, we propose to use an opportunistic forwarding scheme (i.e. a random walk) until the packet reaches a node that is located within the $k$-hop neighborhood of the destination. This is a hybrid algorithm which we refer to as `RANDWLS`. Note that in this paper we do not formally claim that `RANDWLS` is optimal or near-optimal in terms of end-to-end latency for any protocol that admits $k$-hop topology knowledge and 1-hop knowledge of node-availability schedules. However, it is a simple effective protocol that can yield reasonable performance. Hence it is useful to study its performance analytically rather that just using simulations.

In this section we pose the following interesting questions, and answer some of them positively:

1) If a node Z that is $k' > k$ hops away from X sends a packet via random walk, what is the estimated latency suffered by the packet until it *hits* a node that is in the $k$-hop neighborhood of X? (Obviously then the random walk terminates and a shortest path $k$-hop route is followed to reach X.
2) What is the dependence of network properties (size, structure etc.) and the scope of link state dissemination (i.e. value of $k$) on the latency of the aforementioned hybrid routing scheme?
3) What is the fundamental tradeoff between total overhead (link state dissemination and random data forwarding) and end-to-end latency of RANDWLS?

We illustrate the effect of $k$-hop topology information on a random walk by an example in Figure 3. Node 40 wants to send a packet to node 14. In the absence of *any* link state topology dissemination, the random walk would have a propensity to "roam" in the upper half of the topology even after it reaches node 3. However, due to 2-hop dissemination of link state knowledge by node 14, as soon as the random walk reaches either node 5 or node 19, shortest path routing takes over, and thus significantly reduces the latency of the packet.

While the benefits of using limited topology information are intuitively obvious, an analytical characterization of the latency as a function of various network and protocol parameters is necessary to understand the underlying process better.

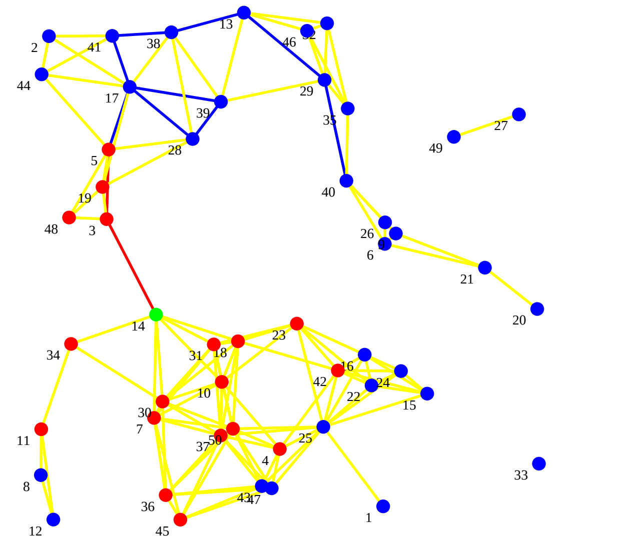

Fig. 3. Random walk issued from node 40 destined for node 14 before it hits the k-hop (k=2) neighborhood of node 14. The blue edges indicate the path taken by the random walk until it hits node 5 which is exactly k hops away from node 14. The red edges indicate the shortest path to the destination from that point onward.

We first show how analysis is possible on a special topology (1-dimensional line lattice) using elementary techniques and then proceed to more advanced techniques for handling general topologies. This analysis will be useful in determining the fundamental tradeoffs between control overhead (i.e. how far should the link state information be spread) and short low-latency paths.

Note that although this simple *hybrid* protocol is by no means optimal, it is however very simple to implement and useful in disruption-prone scenarios where classic stateful protocols break down. This is not just true for proactive link-state protocols – even reactive protocols like DSR and AODV do not work well in disruption-prone networks since stable contemporaneous end-to-end paths are not always present.

### *A. Random walk on line lattice with partial topology knowledge*

Let us consider a finite line lattice containing $N$ nodes, numbered serially from 1 through $N$, such that 1 and $N$ are the two terminal degree-1 nodes. When nodes never go to sleep (i.e. when $p = 1$, or no duty-cycling), the mean hitting time $H_{m,n}$ for a source-destination node pair $\{m, n\}$ is the average number of time-slots a packet needs to traverse from node $m$ to node $n$, when on each time slot the packet randomly moves to a one-hop adjacent node. The expression for $H_{m,n}$ is straightforward to derive using the Lovasz formula [5], and is given by

$$H_{m,n} = (n - m)(m + n - 2), \tag{5}$$

where $1 \le m \le n \le N$. This result can be generalized to the case of opportunistic forwarding [2] with pseudo-random duty-cycling with an average node-awake probability $p$, using

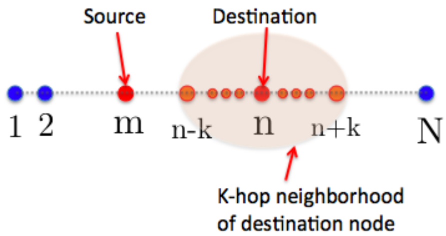


Fig. 4. A line lattice with $N$ nodes. The source and destination nodes are numbered $m$ and $n$ respectively, with $m < n$. The $k$-hop neighborhood is shown using the shaded area, which contains nodes in $[n-k, n+k]$. Under the $k$-hop topology-knowledge model, as soon as the packet hits the $k$-hop boundary of the destination node $n$ (in this case node $n-k$), the packet knows the exact location of $n$, and travels to $n$ hop by hop, taking on an average $k/p$ time slots where $p$ is the pseudo-random duty-cycling rate.

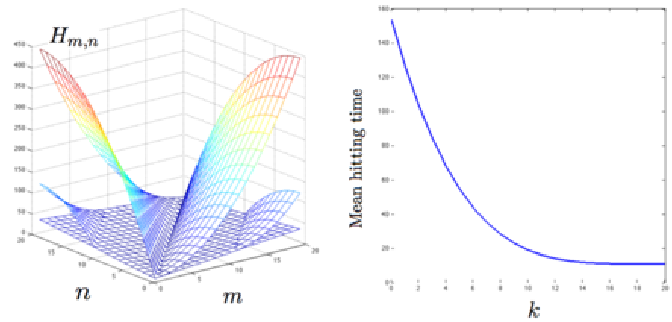


Fig. 5. The plot on the left shows the dependence of mean hitting time $H_{m,n}$ as a function of $m$ and $n$ – the indices of the source and destination nodes respectively, on a line lattice containing $N = 20$ nodes, for $k$-hop topology knowledge with a hybrid routing scheme as described in the paper. The floor of the 3D plot corresponds to full topology knowledge ($k = 20$). The two other contours correspond to $k = 10$, and $k = 0$ (no topology knowledge). The figure on the right plots the average hitting time $\langle H_{m,n} \rangle$ as a function of $k$ when averaged over a uniformly random choice of $m$ and $n$ across the network.

our results in [3]. The resulting hitting time expression is given by

$$H_{m,n} = (n-m)\left[\frac{1}{p} + \frac{m+n-3}{1-(1-p)^2}\right], \tag{6}$$

where $1 \leq m \leq n \leq N$, and $0 < p \leq 1$. $\frac{1}{p}$ and $\frac{1}{(1-(1-p)^2}$ are waiting times (in number of slots) at a node of degree 1 and 2 respectively under pseudo-random duty cycling. Note that Eq. (6) reduces to Eq. (5) for $p = 1$. It is also worth noting that neither of Eqs. (5) or (6) are functions of $N$, because the random walk never goes to the "right" of the destination node $n$. With this observation, an alternative and more intuitive way to obtain the hitting-time expression in Eq. (6) is to solve the tridiagonal system of recursive equations given by:

$$H_{1,n} = \frac{1}{p} + H_{2,n}, \tag{7}$$

$$H_{2,n} = \frac{1}{p} + \frac{1}{2}\left(H_{1,n} + H_{3,n}\right), \tag{8}$$

$$\cdots$$

$$H_{n-2,n} = \frac{1}{p} + \frac{1}{2}\left(H_{n-3,n} + H_{n-1,n}\right), \tag{9}$$

$$H_{n-1,n} = \frac{1}{p} + \frac{1}{2}H_{n-2,n}, \tag{10}$$

with the trivial condition, $H_{k,k} = 0$, $\forall k$. For the explicit solution of the above recursion, see Appendix VI.

Now let us consider the case as depicted in Fig. 4. Let us consider a hybrid routing scheme where a destination node $n$ disseminates topology information to nodes located within its limited scope ($k$-hops, where $k$ is a constant), and a source node $m$ initiates a random walk until the packet finds any node that has received a topology update from the destination. Let us take a source-destination pair $m, n$ with $m < n$. If $m$ is within the $k$-hop boundary of $n$, i.e. $n - k \leq m \leq n$ (the shaded region in Fig. 4), then the mean hitting time $H_{m,n} = \frac{n-m}{p}$ is just the length of the shortest path $(n-m)$ times the mean latency at each hop before the next neighbor along the shortest path comes awake, i.e., $1/p$. If $m$ is outside the $k$-hop boundary of the destination, i.e., $m < n - k$, then the mean hitting time is obtained from Eq. (6) by replacing $n$ with $n - k$ (the mean number of hops to hit the $k$-hop boundary node $n - k$) plus $\frac{k}{p}$, i.e.,

$$H_{m,n} = \frac{k}{p} + (n-k-m)\left[\frac{1}{p} + \frac{m+n-k-3}{1-(1-p)^2}\right] \tag{11}$$

Using the symmetry condition to allow for $m > n$, i.e., $H_{m,n} = H_{N-m+1,N-n+1}$, the mean hitting time can thus be represented in compact notation as follows:

$$\begin{aligned} H_{m,n} &= \frac{\min(k, |n-m|)}{p} \\ &+ \left\{\frac{1}{p} + \frac{(N-k-2) + (N+1-m-n)\text{sgn}(m-n)}{1-(1-p)^2}\right\} \\ &\times \max(0, |n-m| - k), \end{aligned} \tag{12}$$

where $\text{sgn}(x)$ is the sign of $x$, and $1 \leq \{m, n\} \leq N$. We plot in Fig. 5 the expression for hitting time as a function of $m$ and $n$ for $k = 0, 10$ and $20$ for a $N = 20$ node line topology.

### *B. Random walk in general topologies with partial topology knowledge*

In this section, we analyze the performance of RANDWLS in a general network topology which keeps changing due to nodes dynamically duty cycling themselves with probability $p$, and topology link state information is disseminated to only within $k$ hops of each node.

One fact that makes RANDWLS more interesting in the general topology scenario than the one dimensional scenario is the following: in the pure opportunistic forwarding scenario (as we have considered in our prior work [2], [3]), the random walk does not terminate until it reaches a single node – i.e., the destination node. Also, in case of the 1-D scenario in Section III-A, the random walk terminates only when it reaches a single node located at the $k$-hop boundary between the source and the destination. However, in a more general

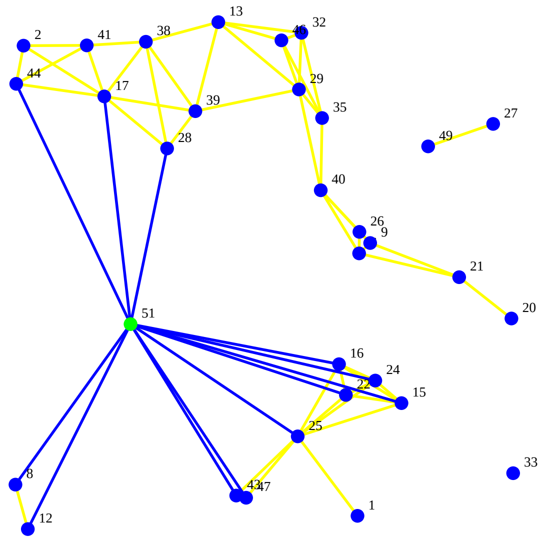


Fig. 6. k-hop neighborhood replaced by a single node

topology, the set of nodes located in the $k$-hop boundary of the destination can have cardinality greater than one. This can be observed from Figure 3. Therefore, a random walk has a finite probability of being terminated at more than one node after it leaves the source. This has the potential of significantly improving the overall latency performance of RANDWLS because the expected time to hit *any* of the $k$-hop neighborhood nodes is intuitively expected to be lower than the single node case.

It is difficult to extend the analytical framework used in Section III-A to suit the general scenario. Hence we use techniques from spectral graph theory after performing the following operations on the original graph.

1) Let $G = (V, E)$ be the original graph and $s, d \in V$ be the source and destination nodes respectively.
2) Define $\Gamma_d^k$ denote the nodes in the $k$-hop neighborhood of node $d$ including the latter.
3) Define $\Delta_d^k$ be the set of nodes in the $k$-hop boundary of $d$. Clearly, $\Delta_d^k = \Gamma_d^{k+1} \setminus \Gamma_d^k$. Let $E_k = \{e = (u, v) | e \in E, u \in \Delta_d^{k+1}, v \in \Delta_d^k\}$.
4) Replace nodes in $\Gamma_v^k$ with a terminal node $t$. Replace edges $e_k \in E_k$ with a set of edges $E_k' = \{e' = (u, t) | u \in \Delta_d^{k+1}\}$.
5) Assign weights to edges in $E_k'$ as follows. For node $u \in \Delta_d^{k+1}$, let $E_u^k$ be the set of edges between $u$ and the nodes in $\Delta_d^k$. Then the weights of these new edges is given by, $w_{(u,t)} = |E_u^k|$.

Figure 6 illustrates the above process when applied to the setup shown earlier in Figure 3. In this example $t = 51$ is the new terminal node and the blue edges are the new edges constructed between the $k+1$-hop boundary of the original destination and node $t$. All older edges $e$ still have weight $w_e = 1$. It is easy to observe that after this construction, the mean latency of a simple random walk between node $s$ and *any* node in the $k$-hop boundary of $d$ is the same as the mean latency of a weighted random walk between $s$ and $t$ in the transformed graph $G'$.

Now, the mean latency between $s$ and $t$ can be computed using results from spectral graph theory [3]. Specifically, the hitting time of a weighted random walk between nodes $u$ and $v$ on a graph $G$ is given by,

$$\mathbf{H}_{u,v} = \sum_{k:\sigma_k \neq 0} \frac{\tilde{W}}{\sigma_k} \Big( \frac{\boldsymbol{\mu}_{k,v}^2}{w_v \ell_v} - \frac{\boldsymbol{\mu}_{k,u} \boldsymbol{\mu}_{k,v}}{\sqrt{w_u \ell_u w_v \ell_v}} \Big) \quad (13)$$

where $w_{u,v}$ is the weight assigned to edge $(u, v)$ that dictates the probability that a random walk goes from $u$ to $v$; this probability is given by $\rho_{u,v} \triangleq \frac{w_{u,v}}{w_u}$; $w_u \triangleq \sum_{v' \in \mathcal{N}_u} w_{u,v'}$ where $\mathcal{N}_u$ is the neighborhood of $u$; $\ell_{u,v}$ is the expected sojourn time for a packet to traverse an edge $(u, v)$; $\ell_u \triangleq \sum_{w \in \mathcal{N}_u} \rho_{u,w} \cdot \ell_{u,w}$ is the expected sojourn time at $u$; $\tilde{W} \triangleq \sum_{(u,v) \in \mathcal{E}} w_{u,v}(\ell_{u,v} + \ell_{v,u}) = \sum_{u' \in \mathcal{V}} w_{u'} \ell_{u'}$; and $\sigma_k$ and $\boldsymbol{\mu}_k$ are the $(k+1)$-th eigenvalue and the corresponding eigenvector of $\tilde{\mathbf{L}}$, where $\tilde{\mathbf{L}}$ is the generalized Laplacian of $G$ with edge weights and sojourn times. Note that we do not describe these terms in detail in this paper. Please refer to our prior work for a complete derivation [3].

Finally, the expected end-to-end latency of RANDWLS from $s$ to $d$ in graph $G$ can be computed by incorporating both random walk and shortest path components as follows:

$$E[L_{s \to t}(G)] = H_{s,t}(G') + \frac{k}{p} \quad (14)$$

While closed form expressions for hitting times do not exist for general graphs, Equation 14 is an *exact* calculation and is not a approximation. Computing hitting times takes $\mathcal{O}(|V|^3)$ running time due to the computation of the eigenvalues and eigenvectors but it is still an analytical means of computing latency performance in contrast with simulation based approaches. We however, do not undermine the power of simulation since certain realistic factors such as channel properties and protocol inefficiencies are more difficult to model analytically. In Section IV we compare how our analytical results compare with simulations.

## IV. Performance Evaluation

In this section we evaluate the RANDWLS routing scheme in general topologies. In particular, we chose to use the standard random geometric graph (RGG) model for wireless multihop scenarios. An RGG is represented by $G(n, r)$ where $n$ is the number of nodes in the graph whose coordinates are drawn from within a two-dimensional unit square, and $r$ is the transmission radius of each node. There is a link between $u$ and $v$ if and only if their coordinates are less than or equal to $r$ distance apart. Admittedly, RGG is too simplistic a model for wireless multihop networks but we chose to use it because it is the simplest model that succinctly captures the spatial aspect of multihop wireless networks, and here it is our endeavor is to understand the basic properties of RANDWLS under

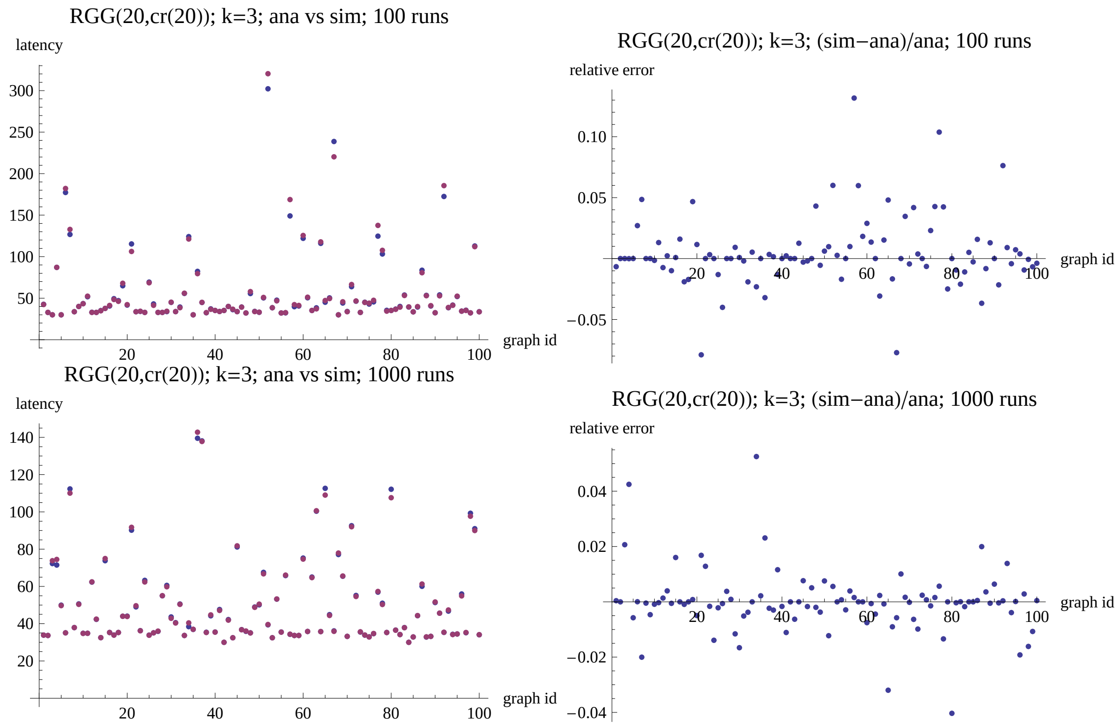


Fig. 7. Comparison between analytical and simulation results for a 20 node random geometric graph at the critical connectivity radius.

perfect communication channels. Performance under more sophisticated channel models is a topic of future research.

To verify the accuracy of the analytical expressions computed in this paper, we compare the results to analysis to results obtained from simulation of RANDWLS. The simulation was performed in a custom random-walk-on-graph simulator with perfect channel conditions and where one packet was sent between the farthest pair of source/destination nodes.

Figure 7 establishes the accuracy of the analytical expressions computed in Section III-B. We simulate a random walk on 100 different RGGs on 20 nodes at the critical radius of connectivity and for duty cycling probability $p = 0.1$. The critical radius of connectivity is given by $cr(n) \approx \sqrt{\frac{\log n}{n}}$. For each instance of an RGG we plotted the latency in number of timeslots between the farthest s-d pair. Both simulation and analysis results are plotted in the two plots on the left. The top left plot shows a scenario where 100 different random walks were simulated between the farthest s-d pair for each of the 100 graphs, whereas the bottom left plot shows the case where 1000 random walks were simulated. The blue points indicate simulation results averaged over the specified number of runs whereas the red dots indicate the analytical results as predicted by Equation 14.

The plots on the right of Figure 7 indicate the deviation between the simulation and analytical results. We observe that simulation results agree closely with analysis and as we average over a greater number of simulation runs, the error reduces significantly, i.e. from mostly within $5\%$ (when averaged over 100 random walks) to mostly within $2\%$ (when averaged over 1000 random walks). The error goes down further when we average over an even larger number of random walks. We do not show results for other topologies such as grid but simulation and analysis agree very closely for those topologies as well.

Now we investigate how the latency of RANDWLS varies as a function of $k$ and $p$. In Figure 8 we study the latency of RANDWLS between the farthest s-d pair in a 100 node RGG shown on the left. We can make two observations from this:

1) For each fixed value of $k$, we see that reducing $p$ has some effect on latency reduction but the gains are marginal beyond a certain value of $p$. However, for sparser topologies such as grids, the latency curve does not flatten as rapidly with increasing $p$.
2) For each fixed value of $p$, we see a significant reduction in latency as $k$ is increased from 0 to 2.

The second observation indicates that with as little topology knowledge as the 2-hop neighborhood in a graph with diameter 8 hops, we can achieve latency results almost as low as what store-and-forward routing on a shortest path would yield.

Finally we study how the mean maximum latency varies in conjunction with topology dissemination overhead in such random networks as a function of scope $k$. We focus on the $p = 1.0$ scenario since this is applies to general routing and not just duty cycling. We do not have an analytical mechanism for computing overhead, hence it is computed as follows:

1) Given graph $G = (V, E)$, for each node $u$, compute the subgraph $G_v(k)$ induced by the vertices in the $k$ hop neighborhood of $u$.
2) Count the number of edges in $G_v(k)$

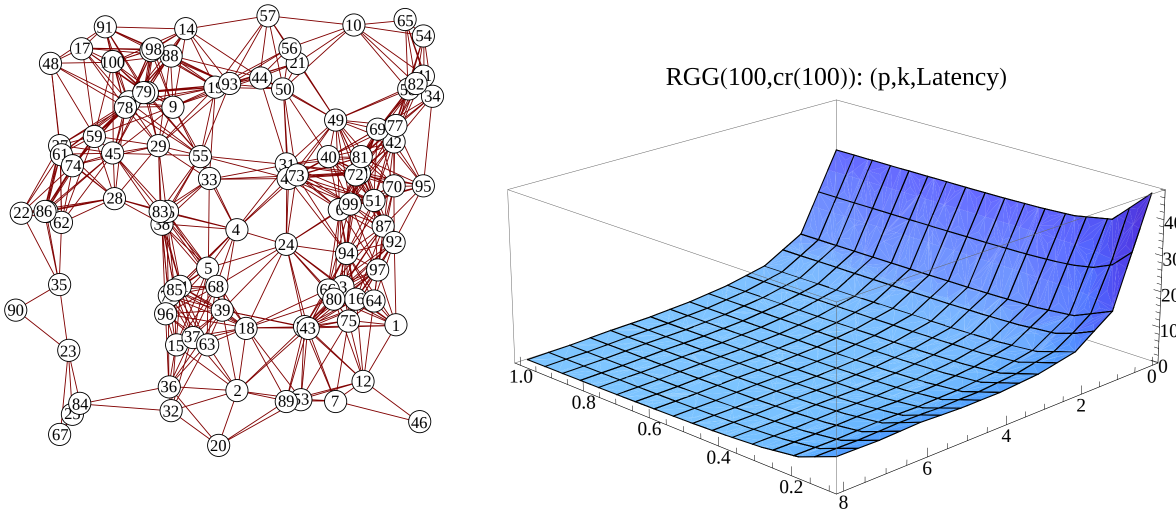


Fig. 8. : A 100 node random geometric graph at the critical connectivity radius, and the analytical dependence of end-to-end latency between the farthest source-destination pair when using the hybrid random-walk/link state hybrid protocol as a function of duty-cycling probability $p$ and dissemination scope $k$.

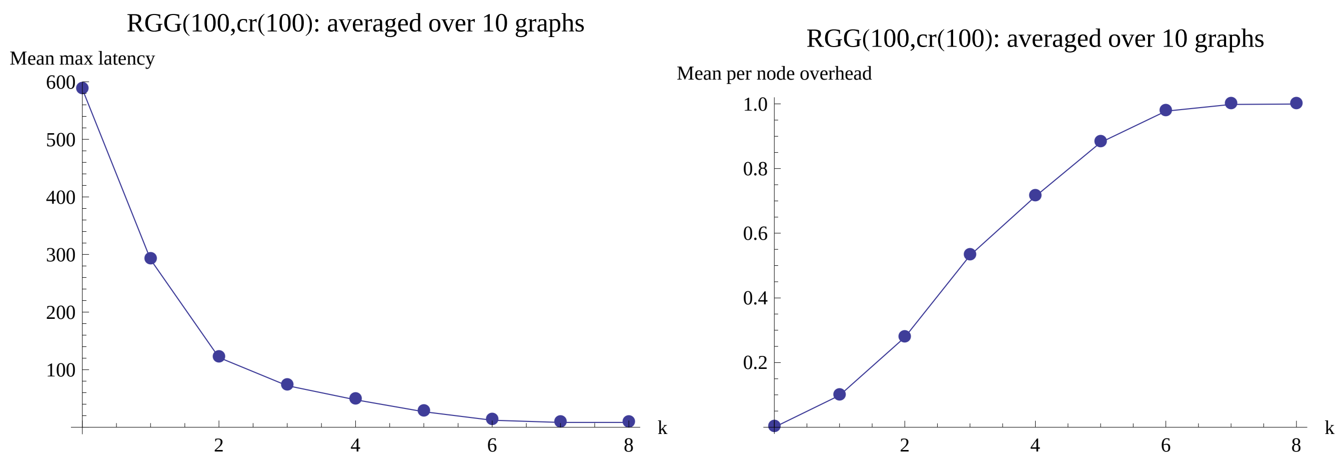


Fig. 9. Latency vs overhead analysis for random geometric graph G(100,cr(100) as a function of topology dissemination scope $k$ for $p = 1.0$: (a) Mean latency as computed from Equation 14; (b) Mean normalized overhead computed per node. The results are averaged over 10 instances of RGGs with the same parameters.

3) The normalized mean overhead per node is given by $Ohd(G,k) = \sum_{v \in V} G_v(k)/(|V||E|)$.

We plot both metrics computed (and averaged) over 10 random instances of RGGs with $n = 100$ and $r = cr(100)$ in Figure 9 side-by-side. It can be observed that as the dissemination scope $k$ goes up from 0 (pure random walk) to 8 (pure shortest path), the mean latency between the farthest pair of nodes drops significantly and $Ohd(k)$ increases with $k$. A key observation from Figure 9 is that both the latency and overhead curves have an inflection point at $k = 2$, i.e., the latency drop becomes more gradual after that and the overhead increase becomes slightly sharper. This indicates that $k = 2$ is a good value to operate the protocol at. Ideally, if we can find a general closed form expression (even approximately) for Equation 14 for random geometric graphs at critical connectivity radius, then these inflection points could be found analytically, if they exist. This is a topic for future research.

Note that such a closed form expression is not difficult to derive for a special topology such as a ring. This is because random walk latency on a ring has a simple closed form expression [2], and collapsing the $k$-hop neighborhood of a node by performing the operation proposed in Section III-B preserves the ring topology; thus the same mathematics can be applied after replacing $n$ with $n - 2k + 1$. This is not true for most other graphs.

## V. Conclusion

In this paper we studied the effect of topology knowledge on the latency performance of an opportunistic forwarding protocol such as random walk. We first derived expressions

$$A^{-1} = \left[\begin{array}{cccccccccc} n-1 & 2(n-2) & 2(n-3) & \ldots & \ldots & \ldots & 8 & 6 & 4 & 2 \\ n-2 & 2(n-2) & 2(n-3) & \ldots & & & & 6 & 4 & 2 \\ n-3 & 2(n-3) & 2(n-3) & \ldots & & & & 6 & 4 & 2 \\ \vdots & \vdots & \vdots & & & & & \vdots & \vdots & \vdots \\ \mathbf{n-m} & \mathbf{2(n-m)} & \mathbf{2(n-m)} & \ldots & \mathbf{2(n-m)} & \mathbf{2(n-m-1)} & \ldots & \mathbf{6} & \mathbf{4} & \mathbf{2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & 2(n-m-1) & \ldots & 6 & 4 & 2 \\ 2 & 4 & 4 & \ldots & & & & 4 & 4 & 2 \\ 1 & 2 & 2 & \ldots & 2 & 2 & \ldots & 2 & 2 & 2 \end{array}\right].$$

Fig. 10. Inverse of the tridiagonal matrix A

for end-to-end latency in a Manhattan routing scenario. We then proposed RANDWLS, a hybrid routing protocol that performs limited-scope dissemination of link-state updates and random walks a packet from the source until it hits any node that has shortest path routing information about the destination. We gave an exact analysis for the mean latency performance of this protocol and observed that for random geometric graphs at the critical connectivity radius, the latency drops sharply as the dissemination scope is increased. This is intuitively expected because the random walk has a greater probability of quickly hitting a node with topology knowledge about the destination. At the same time the overhead due to topology dissemination starts increasing sharply beyond a certain initial scope (it flattens out at the end due to boundary effects). Such analysis gives us a valuable tool to make these predictions systematically without performing time-consuming simulations.

In this paper we only studied the scenario where nodes only know their neighbor's schedules. Presumably if nodes know more information (e.g., schedules of $k$-hop neighbors), then the latency could be further reduced. Also, we only studied the single packet case where the principal source of latency is waiting for a node to come up and the routing latency. Other sources of latency such as congestion are interesting topics for future research.

## References


[1] U. G. Acer, S. Kalyanaraman, and A. A. Abouzeid, "Weak state routing for large scale dynamic networks," *In Proceedings of ACM MobiCom '07*, Montral, Qubec, Canada, September 2007.

[2] P. Basu and C.-K. Chau, "Opportunistic Forwarding in Wireless Networks with Duty Cycling", *in Proceedings of the ACM MobiCom Workshop on Challenged Networks (CHANTS 2008)*, San Francisco, CA, September 2008.

[3] C.-K. Chau and P. Basu, "Exact Analysis of Latency of Stateless Opportunistic Forwarding", *in Proceedings of IEEE INFOCOM '09*, Rio de Janeiro, Brazil, April 2009.

[4] O. Dousse, P. Mannersalo, and P. Thiran, "Latency of Wireless Sensor Networks with Uncoordinated Power Saving Mechanisms," *Proc. ACM MobiHoc '04*, Roppongi, Japan, May 2004.

[5] L. Lovasz, "Random Walk on Graphs: A Survey," *Bolyai Society Mathematical Studies, Keszthely (Hungary), Combinatorics, Paul Erdos is Eighty*, vol. 2, pp. 1–46. 1993.

[6] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks," *Proc. IEEE Infocom*, 2005.

[7] J. Redi, S. Kolek, K. Manning, C. Partridge, R. Rosales-Hain, R. Ramanathan, and I. Castineyra, "JAVeLEN: An Ultra-Low Energy Ad hoc Wireless Network," *Ad Hoc Networks Journal*, Vol. 5, No. 8, December 2007.

[8] C. Santivanez, R. Ramanathan, and I. Stavrakakis, "Making link-state routing scale for ad hoc networks," *In Proceedings of ACM Mobihoc '01*, Long Beach, CA, USA, October 2001.

[9] W. Ye, J. Heidemann, D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, pp. 493–506, Vol. 12, No. 3, June 2004.


## VI. Appendix

Consider the system of equations in Eqs. (7)– (10), with $H_{k,k} = 0$ for $1 \leq k \leq n$. These equations can be represented in the following matrix form,

$$A \begin{bmatrix} H_{1,n} \\ H_{2,n} \\ \vdots \\ H_{n-1,n} \end{bmatrix} = \begin{bmatrix} 1/p \\ 1/(1-(1-p)^2) \\ \vdots \\ 1/(1-(1-p)^2) \end{bmatrix}, \quad (15)$$

where $A$ is the $n-1 \times n-1$ invertible tridiagonal matrix:

$$A = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 & 0 \\ -1/2 & 1 & -1/2 & \ldots & 0 & 0 \\ 0 & -1/2 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ & & & & 1 & -1/2 \\ 0 & 0 & 0 & \ldots & -1/2 & 1 \end{bmatrix}.$$

Inverting Eq. (15) yields the solution,

$$\begin{bmatrix} H_{1,n} \\ H_{2,n} \\ \vdots \\ H_{n-1,n} \end{bmatrix} = A^{-1} \begin{bmatrix} 1/p \\ 1/(1-(1-p)^2) \\ \vdots \\ 1/(1-(1-p)^2) \end{bmatrix}, \quad (16)$$

where the matrix inverse $A^{-1}$ is a $n-1 \times n-1$ matrix that can be obtained in an explicit form as shown in Figure 10.

Thus, we read off the solution for $H_{m,n}$ as,

$$\begin{aligned} H_{m,n} &= \frac{n-m}{p} + \frac{\sum_{j=1}^{n-m} 2j + (m-2)(2(n-m))}{1-(1-p)^2} \\ &= \frac{n-m}{p} + \frac{(n-m)(n-m+1) + (m-2)(2(n-m))}{1-(1-p)^2} \\ &= (n-m)\left[\frac{1}{p} + \frac{m+n-3}{1-(1-p)^2}\right], \quad (17) \end{aligned}$$

which concludes the proof.