Estimating End-to-End Delays Under Changing Conditions

Philippe Nain
INRIA Sophia Antipolis
Philippe.Nain@inria.fr

Don Towsley
UMASS Amherst
towsley@cs.umass.edu

Amotz Bar-Noy
City University of New York
amotz@sci.brooklyn.cuny.edu

Prithwish Basu
Raytheon BBN Technologies
pbasu@bbn.com

Matthew P. Johnson
UCLA
mpjohnson@gmail.com

Feng Yu
City University of New York
fyu@gc.cuny.edu

ABSTRACT

We consider the problem of estimating the end-to-end latency of intermittently connected paths in disruption/delay tolerant networks. This is useful when performing source routing, in which a complete path is chosen for a packet to travel from source to destination (when intermediate nodes are really low complexity devices that can only forward packets but cannot perform route computations), or in linear network topologies. While computing the time to traverse such a path may be straightforward in fixed, static networks, doing so becomes much more challenging in dynamic networks, in which the state of an edge in one timeslot (i.e., its presence or absence) is random, and may depend on its state in the previous timeslot. The traversal time is due to both time spent waiting for edges to appear and time spent crossing them once they become available. We compute the expected traversal time (ETT) for a dynamic path in a number of special cases of stochastic edge dynamics models, and for three different edge failure models, culminating in a surprisingly nontrivial yet realistic "hybrid network" setting in which the initial configuration of edge states for the entire path is known. We show that the ETT for this "initial configuration" setting can be computed in quadratic time (as a function of path length), by an algorithm based on probability generating functions. We also give several linear-time upper and lower bounds on the ETT, which we evaluate, along with our ETT algorithm, using numerical simulations.

Categories and Subject Descriptors

F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General; G.3 [**Probability and Statistic**]: Stochastic processes

Keywords

Dynamic Graphs; Traversal Time; Markov chains; Computation; Appoximation Algorithm

1. INTRODUCTION

In a disruption/delay tolerant network (DTN), edges are typically intermittently unavailable. Specifically, the state of an edge in one timeslot (i.e., present or absent) may be random, as well as possibly dependent on its state in the previous timeslot. If a packet is committed towards traversing such a route (as in source routing[1]), it will simply wait in a node's buffer until the next edge in the path becomes usable. Apart from source routing in DTNs (which could be applicable in general topologies), dynamic paths may be observed in naturally occurring linear network topologies such as military convoys, underwater networks, and linear sensor networks, e.g., for monitoring bridges. Edges in these networks exhibit varying levels of intermittence due to a plethora of factors such as relative mobility of vehicles, significant signal propagation loss due to rugged terrain, or multi-path fading and shadowing.

The time spent traversing a path with such intermittent connectivity includes both the time spent crossing the edges along the path and the time spent waiting (in node buffers) for the edges to appear. Unlike in the case of a static path where each edge is always available for use (and thus the path's traversal time is simply the sum of the delays to cross the constituent edges), computing the expected value of this end-to-end traversal time along a dynamic path is not trivial. This is the problem we study in this paper.

We assume a discrete (slotted) model of time, over which edges appear and disappear; the state of an edge (ON (1) or OFF (0)) in one timeslot depends on its state in the previous timeslot. The simplest model of edge dynamics is the one in which each edge is ON independently with a certain probability p (also referred to as the dynamic Erdős-Rényi (ER) random graph model) in each timeslot. p is assumed to be known in advance or learnt, and the current state of a link is observable by sampling or exchanging messages between endpoints of the said edge.

While the random ER dynamics is a basic model for studying dynamic graphs, it does not capture temporal correlation of edge states. A more useful (and powerful) model is the (q, p) dynamic discrete-time Markovian model [11], in which at each discrete time instant, an edge is either *ON* or *OFF*; and may transition between these states, with known probabilities of transitioning from OFF to ON (p) and from ON to OFF (q), respectively.

The expected traversal time (ETT) for a path with n edges depends on n, q, p, the initial edge states, and the transmission delays on the edges. While the ETT can be straightforwardly estimated

---

[1]In source routing, a complete path is chosen for a packet to travel, from source to destination, within a network [8]. An advantage of source routing over dynamic routing is that once a path is chosen *at source*, no routing decisions need be made along the way – this is advantageous in networks with low complexity devices which can only *forward packets* but not *execute complex routing algorithms*.

by simulation, that method suffers from high variance especially at low values of q, p; hence an exact characterization (algorithmic if not analytic) of the ETT is desirable. Our main result is an $O(n^2)$-time algorithm for this problem. This can serve as a valuable tool for exactly computing or at least bounding the expected end-to-end latency in various DTN scenarios as mentioned earlier.

In this paper, we focus on DTNs, e.g., convoys, where nodes can inform the source(s) periodically about changes in the status of its adjacent edges over alternate communication channels – an example of this is a "hybrid" network where each node is equipped with two transceivers: a high data rate / low transmission range, intermittently connected transceiver to carry "data" (e.g., WiFi) and a low data rate / high transmission range, better connected transceiver for "control" (e.g., cellular network). In such a scenario, the source would be interested in estimating the end-to-end latency of an intermittent WiFi path, given the initial WiFi edge states.

The problem of computing the ETT of a specified dynamic Markovian path is surprisingly nontrivial. In fact, a highly restricted special case of the problem $q = 0, p > 0$ reduces to proving that the highest order statistics of an i.i.d. sequence of $\hat{n}$ geometric random variables is given by:

$$\mathbb{E}[\max\{G_1, \ldots, G_{\hat{n}}\}] = \sum_{i=1}^{\hat{n}} \binom{\hat{n}}{i} \frac{(-1)^{i+1}}{1 - (1-p)^i} = \Theta(\log \hat{n}) \tag{1}$$

An analytic solution to the above statistics problem was nontrivial to prove (originally by [15], later simplified by [4]). In the stated special case of the dynamic path, the traversal time is simply the time taken until *all* $\hat{n}$ OFF edges appear because the $n - \hat{n}$ ON edges never disappear ($q = 0$). Each OFF edge's appearance time is an independent geometric rv, hence the ETT is given by Eq. 1. Computing ETT for general q, p is expectedly nontrivial.

Contributions Our main result is an exact $O(n^2)$-time algorithm for computing the ETT of a length-n path in the general (q, p) model, with edges starting in a specified initial configuration (see Corollary 6 of Sec. 4); the algorithm applies to three edge cost models and to three edge failure models (which vary effects of an edge failure occurring *while* the packet is crossing the failing edge—whether such a packet is delayed and if so, whether its transmission can restart from where it left off; see Sec. 2).

We also compute the ETT for the stationary setting (Sec. 3) and provide a number of easy-to-compute $O(n)$-time upper and lower bounds on the ETT, which we evaluate, along with our ETT algorithm, in numerical simulations. We also show that ETT computed with available link states can be significantly different from that computed with the usual steady state assumption.

Challenges and techniques Designing a polynomial-time algorithm for the ETT in the general (q, p) setting with a given *initial* state requires overcoming several challenges. First of all, the ETT of a dynamic path is not simply the sum of n i.i.d. geometrically distributed rv's (corresponding to waiting times at each edge), as is the case when all edges are in steady state. A dynamic programming (DP) algorithm can be given to compute the ETT for a given initial state s in terms of all possible next-timeslot states (including s itself), but there are exponentially many such states and hence subproblems to solve. Another natural strategy is to compute the expected time $ETT[i]$ to reach each node i on the path: $ETT[i+1]$ depends not just on $ETT[i]$, but also on the probability that edge $(i, i+1)$ is present at the moment when node i is reached (that moment *cannot* be assumed to equal $ETT[i]$). The state of $(i, i+1)$ at that point depends on its state at the previous point *and* on the earlier state of $(i-1, i)$. Since these random states are not independent—there is eventually a large but polynomial number of

subproblems—this leads to a complicated DP with an impractical running time $O(n^{11})$, which we do not present.

The ETT can be numerically approximated, assuming an algorithm to compute $\Pr(T = t)$, by $\sum_{t=0}^{\tau} t \Pr(T = t)$ for a large enough constant τ. $\Pr(T = \tau)$ is nonzero for any arbitrarily large τ, however, and we seek an exact solution. We apply probability and moment generating functions (see e.g. [6]) in order to obtain a much faster, quadratic-time algorithm for the exact computation of ETT.

Related work Time-varying graphs [5] are useful in the study of communication networks with intermittent connectivity such as delay-tolerant networks [7]. In Erdős-Rényi random graphs, each of the $\binom{n}{2}$ edges exists independently with probability p. There is a dynamic, time-slotted extension, and a Markovian generalization [3]. Clementi et al. have also studied asymptotic scaling laws for *flooding times* in dynamic Markovian random graphs [2], however they have not studied the exact computation of end-to-end ETT between an S-D pair. Jones et al. propose the metric of Minimum Estimated Expected Delay (MEED) [9] for use in heuristic DTN routing based on observed contact histories. However, our ETT metric measures the delay along a path whose edges follow a more expressive and analytically tractable (q, p) Markovian model [11]. There are a few works on optimizing *routing latency* in (q, p) Markov networks when only local knowledge of edge states is assumed [12, 13]. However, the model-assisted estimation of end-to-end latency under the availability of actual knowledge of initial edge states has not been considered before in the literature.

2. PRELIMINARIES

We begin with basic assumptions and concepts. Time is discrete, measured in time slots. *Time t* refers to the beginning of the timeslot t (numbered from 0).

We consider a path on $n + 1$ nodes (numbered from 0 to n) and n edges (numbered from 1 to n) obeying Markovian dynamics. In this paper, unless otherwise specified, each edge is assumed to obey i.i.d. dynamics.

DEFINITION 1 (MARKOVIAN (q, p) PATH GRAPHS). *At time 0, each edge in the path is in some known state. The state of a given edge in subsequent time slots is governed by a two-state Markov chain whose transition probabilities are given by the probability transition matrix P for each link $i \in E$:*

$$P = \begin{pmatrix} 1 - p & p \\ q & 1 - q \end{pmatrix}$$

with p (resp. q) the transition probability that link i jumps from state 0 (resp. state 1) into state 1 (resp. state 0) in one timeslot.

In the $(1,1)$ setting, edge states alternate deterministically. In the $(1 - p, p)$ setting, an edge's state is independent of its previous state (also known as the Erdős-Renyi setting).

DEFINITION 2 (STEADY STATE PROBABILITIES). *The stationary probabilities that each link i is in states OFF (0) and ON (1) are $\pi_0 = \pi_{off} = \frac{q}{p+q}$ and $\pi_1 = \pi_{on} = \frac{p}{p+q}$, respectively.*

DEFINITION 3 (TRANSIENT PROBABILITIES). *Let $P_{a,b}(i, t) = \Pr(X_i(t) = b \mid X_i(0) = a)$ be the probability that link i is in state $b \in \{0, 1\}$ at time t given that it was in state $a \in \{0, 1\}$ at time $t = 0$ for initial state X. Note that $P_{a,b}(t)$ does not depend on the link's identity since all links are assumed to have the same parameters p and q. Let $\beta = 1 - p - q \in [-1, 1]$. It is known [10] that for any $t \geq 0$:*

$$p_{10} = P_{1,0}(t) = \pi_0(1 - \beta^t); \quad p_{11} = P_{1,1}(t) = \pi_1 + \pi_0\beta^t \tag{2}$$

$$p_{01} = P_{0,1}(t) = \pi_1(1 - \beta^t); \quad p_{00} = P_{0,0}(t) = \pi_0 + \pi_1\beta^t \tag{3}$$

DEFINITION 4 (EDGE TRANSMISSION DELAY). *Edge i has transmission delay d_i, which may be a random variable (rv). Transmission delays are all 0 in the* Cut-Through *(CuT) model, all 1 in the* Store *or* Advance *(SoA) model, and nonnegative integers in the* Distance *(Dist) model. $D = \sum_{i=1}^{n} d_i$.*

Transmission assumptions Edge state indicates whether a packet can *begin* crossing the edge and, depending on the failure model, whether and when it will succeed. If the packet reaches an edge's entry node *when the edge is ON*, then it immediately starts traversing the edge; if the edge is *OFF*, the packet waits there until the edge appears. Edge transmission takes zero or more slots, depending on edge transmission delay. With zero-delay edges (CuT), an unlimited number of contiguous *ON* edges can be crossed instantly (modeling situations in which transmission times are negligible relative to time scales of disruption and repair [14]).

We consider three edge failure models. In all cases, the packet requires a nonnegative number of time slots to cross a given edge, if it is ON. The models differ as to what occurs when the edge fails prior to the transmission's completion:

1. **Unaffected:** Transmission continues while the edge is OFF; it simply cannot start unless the edge is ON.

2. **Resume:** The packet resumes its progress across the edge once the latter turns ON again.

3. **Retransmit:** The packet must be retransmitted on the edge in its entirety from scratch.

Note that the three models are equivalent in both CuT and SoA, but yield different behavior in Dist. Under **resume**, a transmission successfully completes once a total of d_i ON timeslots for edge i occur; under **retransmit**, a transmission completes once d_i ON timeslots for edge i occur in a row.

Notation The *state* or *configuration* of all edges at time t is indicated by $X(t)$, which represents a bitstring (or path) of length n. The state (0 or 1) of a particular edge i at time t is indicated by $X_i(t)$, sometimes written X_i^t when notationally convenient. We use lower-case variables (e.g., x or y) to indicate particular, fixed bitstrings. T is a random variable indicating time of arrival at node n (e.g. t), with $\mathbb{E}[T] = ETT$. We sometimes write $ETT(P|X)$ or $ETT(X)$ to indicate ETT for path P embedded in a graph with initial configuration X, and $ETT[i]$ for the expected time of arrival at node i in the context of a path. We abbreviate Markov chain (MC), upper bound (UB), and lower bound (LB).

3. ETT IN THREE SIMPLE SETTINGS

In this section, we will examine three special case settings, where computing ETT (or even the distribution of traversal time) is relatively simple.

3.1 The deterministic setting

Let the $(q, p) = (1, 1)$, and assume each d_i is a constant. Then the exact traversal time can easily be computed. First, consider the **unaffected** model. Let b_i indicate the initial state of edge $(i-1, i)$.

PROPOSITION 1. *Let $b_0 = 1$ and $d_0 = 0$. Let $\Delta_i = |b_{i+1} - b_i|$ indicate whether $b_{i+1} \neq b_i$, and let $k = \sum_{i=0}^{n-1} \Delta_i$. Then the traversal time under **unaffected** is $D + \sum_{i=0}^{n-1} ((d_i + \Delta_i) \mod 2)$ in Dist, $2n - k + 1$ in SoA, and k in CuT.*

PROOF. The traversal time is the sum of the total time spent crossing edges D and the total time spent waiting. At each edge i, we wait one timeslot in two cases: b_i differs from b_{i-1} and d_{i-1} was even, or $b_i = b_{i-1}$ and d_{i-1} was odd. In SoA, this is $D = n$

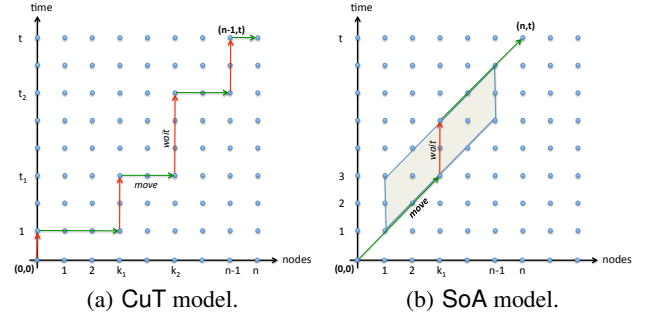(a) CuT model.   (b) SoA model.

Figure 1: Analyzing the latency in Markov path graphs.

plus $\sum_{i=0}^{n-1} ((d_i + \Delta_i) \mod 2)$, which is $\bar{b}_0$ plus the number of positions $i > 0$ where $b_{i+1} = b_i$, or $n + n - k + 1$. In CuT, we have $D = 0$, and $\sum_{i=0}^{n-1} ((d_i + \Delta_i) \mod 2)$ is simply k. $\square$

By reducing Dist to SoA, we obtain:

COROLLARY 2. *The traversal time for Dist under **resume** is $2D - k + 1$.*

PROOF. The **resume** model can be reduced to **unaffected** by modeling each edge of size d_i as a sequence of d_i unit-size edges, all with the same initial state as d_i, for a total of D such edges. Because of the chosen initial states, k in the resulting instance will be the same as k in the original. $\square$

With $d_i > 1$ for some i, the **retransmit** model does not apply to the deterministic setting because the packet will never succeed in crossing d_i.

3.2 The $(1 - p, p)$ stochastic model

Computing the ETT here is straightforward.

PROPOSITION 3. *If the transmission latency for edge $(i-1, i)$ is given by a rv d_i, the ETT under **unaffected** is $\sum_i \mathbb{E}[d_i] + n(1 - p)/p$ in Dist, or $\frac{n}{p}$ and $\frac{n(1-p)}{p}$ in SoA and CuT, respectively.*

PROOF. The total expected transmission time is $\sum_i \mathbb{E}[d_i]$. The expected wait for each edge to appear is $1/p$ if the edge is OFF (with probability $1 - p$) and 0 otherwise. $\square$

We note that when each d_i is constant, the full distribution of end-to-end latency can be computed: $\Pr(T = t)$ for $t \geq D$ is the probability that $t - D$ timeslots are spent waiting at nodes 0 through $n - 1$, analogous to throwing identical balls into bins:

$$\Pr(T = t) = \binom{t - D + n - 1}{t - D} p^n (1 - p)^{t-D} \quad (4)$$

3.3 The (q, p) Markov model in steady state

If each Markov chain has converged (or *mixed*) before transmission begins, we have the steady state probabilities given by Definition 2. As a corollary of Proposition 3 above, we then have:

COROLLARY 4. *If d_i is again an rv, the expected routing time under **unaffected** is $\sum_i \mathbb{E}[d_i] + n(1 - \pi_1)/p$ in Dist, or $n(1 + \frac{1 - \pi_1}{p})$ and $\frac{n(1 - \pi_1)}{p}$ in SoA and CuT.*

Now we show how to compute the full probability distribution of end-to-end latency for this case. Fig. 1 illustrates paths through space-time corresponding to the progress of the packet traveling from node 0 to n, in CuT and SoA. Each such path is composed of segments of *moving* and *waiting*. Let m be the number of waiting segments, each preceded by a moving segment (possibly empty in the first case). Let the path be specified by a sequence (see Fig. 1) $\{0, 0, k_1, t_1, k_2, t_2, \ldots, k_m, t_m, n, t\}$, and assume each d_i is constant. Clearly $0 \leq m \leq \min(n, t - D)$. Since edge transmission takes time d_i, the total latency t obeys $t \geq D = \sum_e d_e$. Notice

that if $t = D$, then there will be no waiting segment, which is $m = 0$. Then the probability of unique path $P = \{(0,0), (n,t)\}$ is $\pi_1^n = (\frac{p}{p+q})^n$.

Now we consider the situation of $t > D$ and $m > 0$. Since the state of an edge i over time is governed by a Markov chain, the probability of a *waiting* segment of length ℓ is $\pi_0 (1-p)^{\ell-1} p$. The probability of some path $P = \{(0,0), (k_1,t_1), (k_2,t_2), \ldots, (k_m,t_m), (n,t)\}$ conditioned on $k_1 = 0$ is given by the following:

$$(\pi_0(1-p)^{t_1-1}p)\pi_1^{k_1-1} \cdot (\pi_0(1-p)^{t_2-t_1-1}p)\pi_1^{k_2-k_1-1} \cdots$$
$$(\pi_0(1-p)^{t-t_m-1}p)\pi_1^{n-k_m-1}$$
$$= \pi_1^{n-m} \pi_0^m (1-p)^{t-D-m} p^m$$
$$= \left(\frac{p}{p+q}\right)^{n-m} \left(\frac{q}{p+q}\right)^m (1-p)^{t-D-m} p^m$$
$$= \frac{p^n q^m (1-p)^{t-D-m}}{(p+q)^n}$$

The probability of path P conditioned on $k_1 > 0$ is the same.

We now count the total number of possible paths with exactly m waiting segments, conditioning on two cases of the value of k_1. In the $k_1 = 0$ case, m waiting segments implies m moving segments. Such a path is generated by placing $m - 1$ breakpoints on the space axis and $m - 1$ breakpoints on the time axis, for a total of $\binom{n-1}{m-1}\binom{t-D-1}{m-1}$ possible such paths. In the $k_1 > 0$ case, m waiting segments implies $m + 1$ moving segments, for a total of $\binom{n-1}{m}\binom{t-D-1}{m-1}$ possible such paths. Together, this yields a total $\binom{n-1}{m-1}\binom{t-D-1}{m-1} + \binom{n-1}{m}\binom{t-D-1}{m-1} = \binom{n}{m}\binom{t-D-1}{m-1}$ possible paths. The full end-to-end latency probability distribution $\Pr(T = t)$ for $p > 0, q > 0$ is then given by:

$$\Pr(T = t) = \sum_{m=1}^{\min\{n, t-D\}} \binom{n}{m}\binom{t-D-1}{m-1}\frac{p^n q^m (1-p)^{t-D-m}}{(p+q)^n} \quad (5)$$

REMARK 1. *In settings 3.2 and 3.3, Proposition 3 and Corollary 4 could be simply extended to the non-uniform (q_i, p_i) case for each edge i, by replacing n by $\sum_{i=1}^n d_i$.*

4. ETT UNDER MARKOV MODEL WITH INITIAL CONFIGURATION

In this section, we give the main result of this paper – an efficient algorithm to compute the ETT of a dynamic Markovian path graph with a given initial configuration of edge states. We use the basic definitions from Section 2. Detailed proofs of the results in this section have been omitted for paucity of space, and can be found in a longer technical report [1].

Let T_i be the time at which the packet reaches link i, i.e. the time at which it finishes crossing link $i - 1$. Let $D_i = T_{i+1} - T_i$ be the time spent waiting for link i to appear plus the time taken to cross it. To compute ETT, i.e., $\mathbb{E}[T_n]$, the main strategy to reduce computational complexity is to maintain two probability generating functions (PGF), one for arrival time T_i, which is general for all failure models:

$$G_{i,x}(z) = \mathbb{E}[z^{T_i} | X(0) = x]$$

for initial edge state $x = (x_1, ..., x_n)$ and $i = n$; and another PGF for the total delay in traversing edge i, i.e., D_i (which includes time to wait for edge i to come up and the transmission delay d_i) conditioned upon the state which the packet found the edge i in when it arrived:

$$F_{i,1}(z) = \mathbb{E}\left[z^{D_i} | X_i(T_{i-1}) = 1\right]$$
$$F_{i,0}(z) = \mathbb{E}\left[z^{D_i} | X_i(T_{i-1}) = 0\right]$$

Also, define for all links $i \in \{1, 2, ..., n\}$,

$$\gamma_{i,1} = \mathbb{E}[D_i | X_i(T_{i-1}) = 1], \quad \gamma_{i,0} = \mathbb{E}[D_i | X_i(T_{i-1}) = 0]$$

Note that $\gamma_{i,1} = d_i$, also $\gamma_{i,1} = dF_{i,1}(z)/dz|_{z=1}$ and $\gamma_{i,0} = dF_{i,0}(z)/dz|_{z=1}$. The edge failure models of Sec. 2 only affect the computation of expressions for $F_{i,0}(z)$ and $F_{i,1}(z)$, which we give below. For any number $a \in [0,1]$ let $\bar{a} = 1 - a$. Note that expressions of the form $a\alpha + \bar{a}\beta$ are equivalent to α **if** $a = 1$ **else** β.

4.1 Computing the PGFs and the ETT

Our computation of the PGFs $G_{i,x}(z)$ of T_i invokes the PGFs $F_{i,b}(\beta^j)$ of D_i, for $b \in \{0,1\}$, and $i \in \{1, \ldots, n\}, j \in \{0, \ldots, n-1\}$. Since $F_{i,b}(\beta^j)$ are computed differently under different failure models, whereas $G_{i,x}(z)$ are not, in this subsection we give a recursive procedure for computing $G_{i,x}(z)$, assuming that $F_{i,b}(\beta^j)$ have been pre-computed. Later, we show how to compute each $F_{i,b}(\beta^j)$ in constant time, for the three failure models.

THEOREM 5 (COMPUTING $G_{i,x}(z)$). *For initial state $X(0) = x = (x_1, ..., x_n)$, we can (given $F_{i,0}(\beta^j)$, $F_{i,1}(\beta^j)$ for $i = 1, ..n$ and $j = 0, ..n - 1$), compute the following in $O(n^2)$ time:*

$$G_{1,x}(z) = x_1 F_{1,1}(z) + \bar{x}_1 F_{1,0}(z)$$
$$G_{i,x}(z) = \phi_i(z) \cdot G_{i-1,x}(z) + \chi(i) \cdot \psi_i(z) \cdot G_{i-1,x}(\beta z),$$
$$i = 2, ..., n, \text{ where:}$$
$$\phi_i(z) = \pi_0 F_{i,0}(z) + \pi_1 F_{i,1}(z),$$
$$\psi_i(z) = F_{i,0}(z) - F_{i,1}(z),$$
$$\chi(i) = \bar{x}_i \pi_1 - x_i \pi_0$$

COROLLARY 6 (COMPUTING ETT). *For initial state $x = (x_1, ..., x_n)$, we can (given the values $F_{i,0}(\beta^j)$, $F_{i,1}(\beta^j)$ for $i = 1, ..., n$ and $j = 0, ..., n - 1$) compute ETT in $O(n^2)$ time:*

$$\mathbb{E}[T_n] = \sum_{i=1}^n \pi_0 \gamma_{i,0} + \pi_1 \gamma_{i,1} + (\gamma_{i,0} - \gamma_{i,1}) \cdot \chi(i) \cdot G_{i-1,x}(\beta) \quad (6)$$

4.2 Computing $F_{i,1}(z)$ and $F_{i,0}(z)$

Let S be the random variable (rv) denoting the time needed to traverse a link and Y the rv denoting the amount of time spent waiting, upon arrival there, for the link to turn on. First observe that for all links $i \in \{1, ..., n\}$:

$$F_{i,0}(z) = G_Y(z)F_{i,1}(z) \quad (7)$$

since $D_i|_{X_i(T_i)=0} =_d Y + D_i|_{X_i(T_i)=1}$, ($=_d$ indicates equality in distribution) where Y is a geometrically distributed rv with parameter p, independent of $D_i|_{X_i(T_i)=1}$. (Recall that a sum of rv$s$ yields a product of PGFs.) In our model, since Y corresponds to the duration of an off period of a link, we have $\Pr(Y = k) = (1-p)^{k-1}p$, which yields the following PGF:

$$G_Y(z) = \frac{pz}{1 - (1-p)z}, \quad 0 \le |z| \le 1 \quad (8)$$

The PGFs of D_i, i.e. $F_{i,1}(z)$ and $F_{i,0}(z)$ are given in Table 1 for each of the three failure models.

Note that higher order moments of T_n can be computed from PGF $G_{n,x}(z)$ but the computational complexity will be much higher.

4

Failure Model	$F_1(z)$	CuT	SoA
Unaffected	$G_S(z)$	1	z
Resume	$\dfrac{G_S(1-q(1-G_Y(z)))-\Pr(S=0)q(1-G_Y(z))}{1-q(1-G_Y(z))}$	1	z
Retransmit	$\dfrac{G_S((1-q)z)-q\Pr(S=0)}{\Pr(S=0)+\Pr(S=1)(1-q)z-G_S((1-q)z)}$	not valid	not valid

Table 1: Computing PGF for edge traversal rv's, D_i (Recall that $G_S(z)=1$ for CuT, z for SoA)

5. STOCHASTIC ANALYSIS AND BOUNDS

In Sec. 4, we showed that the ETT can be computed in quadratic time. In this section (proofs in [1] due to space constraints) we give bounds for ETT in a number of settings, either analytically or by linear-time algorithms. Some of these bounds apply to all three edge cost models (SoA, CuT, Dist) and all three edge failure models, while others apply only to **unaffected** failure model (recall from Sec. 2 that all three edge failure models are equivalent under CuT and SoA), while others are tailored specifically to SoA, as indicated below.

THEOREM 7. (BOUNDS FOR GENERAL $(q, p > 0)$, VIA HOLDER INEQ) *Let x be an an initial state, and let $B_x(k)$ be the indicator function for the set of bit positions equalling 1 in x. Let $SS = S_n + n\pi_0/p$ be the expected cost in steady state. Then for all cost models and failure models, UBs and LBs for $ETT(x)$ are given by:*

$$\left| SS - ETT(x) \right| \leq \frac{1}{p+q} \sum_{k=0}^{n-1} \left(\frac{q}{p}\right)^{B_x(k)} |1-p-q|^{S_k} \quad (9)$$

where $S_k = \sum_{i=1}^{k} \gamma_{i,1} = \sum_{i=1}^{k} d_i$.

REMARK 2. $p_{01}(t)$ *increases and* $p_{11}(t)$ *decreases monotonically in t when $p + q < 1$, indicating stable networks. In contrast, $p + q > 1$ indicates less stable networks.*

PROPOSITION 8. *For fast dynamics ($p+q \geq 1$), for any initial state x in the CuT model, $ETT(x) \geq \frac{n(1-p)}{p}$.*

Stable dynamics ($p + q < 1$): We now develop a stochastic ordering relation on initial states to give some tighter bounds for a commonly occurring case in DTNs, where the network evolves slowly.

Consider initial states $x, y \in \{0,1\}^n$. Then y stochastically dominates x (denoted by $x \prec y$) if $\forall_k : \sum_{i=1}^{k} x_i \geq \sum_{i=1}^{k} y_i$.

Let U, V be positive random variables with CDFs $P_U(u) = \Pr(U \leq u)$ and $P_V(v) = \Pr(V \leq v)$. We say that V is larger than U in distribution (written $U \leq_d V$) if $\mathbb{E}[f(U)] \leq \mathbb{E}[f(V)]$ for all increasing functions f. It is also the case that $U \leq_d V$ iff $\forall_u : P_U(u) \geq P_V(u)$.

Let $T_i(x)$ be the random variable denoting the time taken to traverse the first i edges in the given initial configuration $x \in \{0,1\}^n$, and $T(x) = T_n(x)$. Then we have the following Lemma, Corollary, and Theorem.

LEMMA 1. *Let $x, y \in \{0,1\}^n$ be two edge state vectors that are identical in all but the first two positions where $x_1 = y_2 = 1$ and $x_2 = y_1 = 0$. Then $T(x) \leq_d T(y)$ under the **unaffected** failure model with all three edge cost models.*

COROLLARY 9. $T_i(x) \leq_d T_i(y)$ *if $x \prec y$, for $i = 1, \dots, n$.*

THEOREM 10. *If $x, y \in \{0,1\}^n$ are two state vectors such that $x \prec y$, and $p + q < 1$, then $T(x) \leq_d T(y)$ under the **unaffected** failure model with all three cost models.*

It is easy to see that for all edge cost models, all 2^n possible initial configurations form a partial order with respect to expected traversal time, with 0^n and 1^n denoting the greatest and least elements of the poset respectively. In particular, we have:

COROLLARY 11. *For a given initial state x with k 0s and $n - k$ 1s, for $p + q < 1$, $ETT(1^n) \leq ETT(1^{n-k}0^k) \leq ETT(x) \leq ETT(0^k 1^{n-k}) \leq ETT(0^n)$.*

The next two corollaries state tighter bounds for the SoA model for some specific initial configurations.

COROLLARY 12. *For $p + q < 1$ and SoA model, it follows from Theorem 7 that:*

$$\left| n\left(1 + \frac{q}{p(p+q)}\right) - ETT(0^n) \right| \leq \frac{1 - (1-p-q)^n}{(p+q)^2} \quad (10)$$

The LB on $ETT(0^n)$ and the UB on $ETT(1^n)$ can be tightened for $p + q < 1$ by Remark 2 with respect to steady state:

COROLLARY 13. *For the SoA model, $ETT(1^n) \leq n\left(1 + \frac{q}{p(p+q)}\right) \leq ETT(0^n)$.*

Following is another configurable bound based on the fact that Markov chains eventually reach steady state.

THEOREM 14 (MC STEADY STATE). *For $p + q < 1$ in the CuT and SoA models, initial configuration x, and an appropriate MC convergence threshold ϵ:*

$$ETT(x) \leq t_0^* + (n - k^*)\left(1 + \frac{q}{p(p+q)}\right)$$

$$\text{where } t_0^* > \frac{\log \epsilon + \log(p+q) - \log p}{\log(1-p-q)} \quad (11)$$

and $k^* \in [1, n]$ is the smallest integer that satisfies:

$$k^*\left(1 + \frac{q}{p(p+q)}\right) + \frac{1 - (1-p-q)^{k^*}}{(p+q)^2} \geq t_0^*$$

Tighter bounds So far we have shown relatively simple bounds. We now demonstrate a bounding technique which leverages the fact that $p + q < 1$ (and thus the monotonicity of $p_{10}(t), p_{00}(t)$), and is later combined with the techniques presented earlier to get much tighter bounds.

THEOREM 15 (VIA JENSEN'S INEQ). *If T_k^u is an upper bound for $ETT(1^k)$ and T_k^l is a lower bound for $ETT(0^k)$, then the following[2] hold when $p + q < 1$ (for all three edge cost models and all three transmission failure models):*

$$ETT(1^n) \leq T_{n-1}^u + \gamma_{n,1} + p_{10}(T_{n-1}^u)/p$$
$$ETT(0^n) \geq T_{n-1}^l + \gamma_{n,1} + p_{00}(T_{n-1}^l)/p \quad (12)$$

COROLLARY 16. *Let T_x^u and T_x^l be upper and lower bounds, respectively, for the configuration x traversed so far. Then (for all edge cost and failure models) we can compute the following bounds iteratively:*

$$\mathbb{E}[T_i(x1)] \leq T_s^u + \gamma_{n,1} + p_{10}(T_x^u)/p \quad (13)$$
$$\mathbb{E}[T_i(x0)] \geq T_s^l + \gamma_{n,0} + p_{00}(T_x^l)/p \quad (14)$$

Iterative combining of bounds For arbitrary initial states x, the above Jensen's bounds and (9) can be combined iteratively to bound $ETT(x)$. To compute the UB in Corollary 11, one can use the UBs for $ETT(0^k)$ and then numerically compute $ETT(0^k 1)$, $ETT(0^k 11)$,

---
[2] Recall from Sec. 4 that $\gamma_{n,1} = \mathbb{E}[D_n \mid X_n(T_{n-1}) = 1]$.

(a) All edges down. **(b) Random 25% edges up.** **(c) Random 75% edges up.**
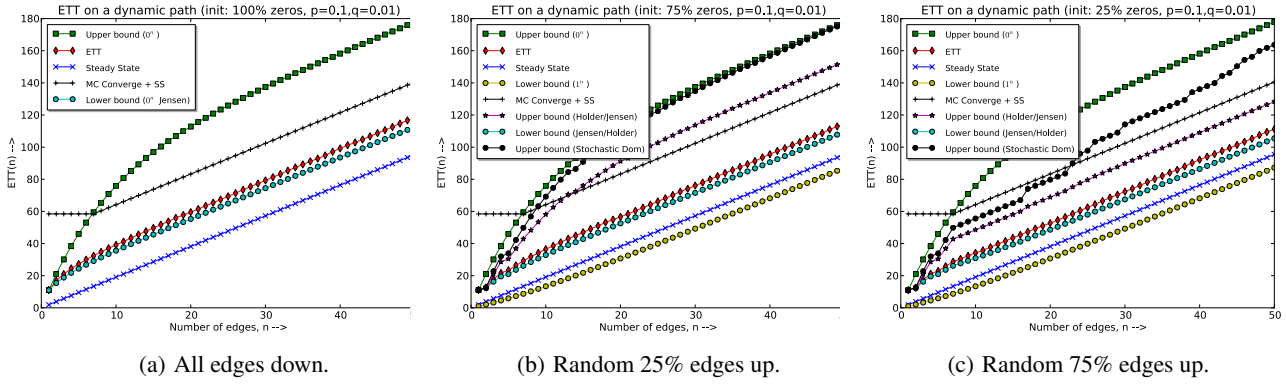
Figure 2: Comparison of various bounds for low p and q ($p = 0.1, q = 0.01$) compared against the actual ETT (for the SoA model), computed as follows: *Upper bound* (0^n) **via (10).** *Steady state* **via Corollary 13.** *Lower bound* (1^n) **via (9).** *Upper bound* $(Holder/Jensen)$ **via iterative application of the upper bound of (9) upon encountering a 0 and Jensen's ineq. upon encountering a 1. Similarly,** *Lower bound* $(Jensen/Holder)$ **via iterative application of Jensen's ineq. upon encountering a 0 and the lower bound of (9) upon encountering a 1.** *Upper bound (Stochastic Dom)* **via the strategy outlined in Corollary 11.** *MC Converge + SS* **via Theorem 14. For (a),** *Lower bound* $(0^n$ *Jensen)* **via Theorem 15.**

..., $\text{ETT}(0^k 1^{n-k})$ by iterative application of (13). Similarly, for computing the LB, one can use the LBs for $\text{ETT}(1^{n-k})$ (i.e., the LB of (9)) and then numerically compute $\text{ETT}(1^{n-k}0)$, $\text{ETT}(1^{n-k}00)$, ..., $\text{ETT}(1^{n-k}0^k)$ by iterative application of (14).

Validation We implemented the ETT algorithm and computed the bounds on randomly generated paths, for various values of p, q and initial configurations (see Fig. 2). We also note that for moderate and high values of p and q, the bounds are accurate and often overlap with the ETT curve; hence we are only highlighting a specific low duty cycle scenario ($p = 0.1, q = 0.01$) where the differences between them and the ETT curve become apparent. In general, it can be observed from all four illustrated scenarios that while the iteratively computed lower bound performs quite well, there is room for improvement in the upper bounds. We also observe that the steady state bounds that tend to ignore the actual edge states do not perform well, thus vindicating our decision to take the initial configuration into account for computing ETT.

6. DISCUSSION AND FUTURE WORK

In this paper we (efficiently) computed exactly and bounded the expected time to traverse a dynamic path with known edge states whose future evolution is governed by Markov chains. This is useful in source routing in networks with many computationally challenged devices and in several DTNs with linear topologies.

Natural interesting generalizations include edge dynamics that are not independent. For example, adjacent pairs of edge failures could correspond to a node failure, and probabilities q_i, p_i that vary by edge. Our algorithms maintain the same complexity as long as the sum $p_i + q_i$ is held constant for all i, but in the full generalization become exponential-time.

7. REFERENCES

[1] A. Bar-Noy, P. Basu, M. Johnson, P. Nain, D. Towsley, and F. Yu. Estimating Durations Under Changing Conditions. Technical report, Raytheon BBN Technologies, January 2012. Available at http://www.ir.bbn.com/~pbasu/pubs/markovpath.pdf.

[2] A. E. F. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri. Flooding time in edge-markovian dynamic graphs. In *PODC*, Toronto, Canada, August 2008.

[3] A. E. F. Clementi, A. Monti, F. Pasquale, and R. Silvestri. Communication in dynamic radio networks. In *PODC*, Portland, OR, August 2007.

[4] B. Eisenberg. On the expectation of the maximum of iid geometric random variables. *Statistics and Probability Letters*, 78:135–143, 2008.

[5] A. Ferreira. Building a reference combinatorial model for manets. *IEEE Network*, 18(5):24–29, 2004.

[6] G. Grimmett and D. Welsh. *Probability: An Introduction*. Oxford, 1986.

[7] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM*, pages 145–158, New York, NY, USA, 2004.

[8] D. B. Johnson, D. A. Maltz, and J. Broch. *DSR: the dynamic source routing protocol for multihop wireless ad hoc networks*, pages 139–172. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, December 2000.

[9] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical routing in delay-tolerant networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, pages 237–243, New York, NY, USA, 2005. ACM.

[10] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[11] V. Manfredi, M. Crovella, and J. Kurose. Understanding stateful vs stateless communication strategies for ad hoc networks. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 313–324, New York, NY, USA, 2011. ACM.

[12] E. Nikolova and D. R. Karger. Route planning under uncertainty: the canadian traveller problem. In *AAAI*, pages 969–974, 2008.

[13] R. G. Ogier and V. Rutenburg. Minimum-expected-delay alternate routing. In *INFOCOM*, pages 617–625, 1992.

[14] R. Ramanathan. Challenges: A radically new architecture for next generation mobile ad hoc networks. In *Proc. of MOBICOM*, Cologne, Germany, August 2005.

[15] W. Szpankowski and V. Rego. Yet another application of binomial recurrence: order statistics. *Computing*, 43:401–410, 1990.