

# Message Fragmentation for a Chain of Disrupted Links

Philip Ginzboorg  
Nokia Research Center, Finland  
philip.ginzboorg@nokia.com

Valtteri Niemi  
Nokia Research Center and  
Turku University, Finland  
valtteri.niemi@nokia.com

Jörg Ott  
Aalto University, Finland  
jo@netlab.tkk.fi

**Abstract**—We investigate the problem of estimating the transmission time of fragmented messages over multiple disrupted links. We build a system model for the case where a single message is sent over a chain of links and the disruptions in these links are identically and independently distributed. For this case, we derive an approximation formula for the mean transmission time, based on number of links, length of fragments and distributions of disruptions. The formula is verified against simulation experiments in the cases of uniform and exponential distributions for disruptions.

## I. INTRODUCTION

Delay-tolerant networks (DTNs) [1] may use potentially large messages (rather than small packets) as basic transmission unit offered to applications. Sending large messages implies that those will be broken down into individual packets for the actual transmission across a physical link that comply with the link's MTU size. Such a mechanism is defined, e.g., for the convergence layers of the DTN bundle protocol [2].

Since messages may be large, their transmission as a series of packets may not complete during a *contact* period, i.e., while a link is available ("ON"). When a link comes up again after a down ("OFF") period (the *inter-contact time*), the message transmission should resume (roughly) where it stopped, rather than have to restart from the beginning. For this purpose, it is required to *fragment* a message into smaller pieces ("units") whose transmission is more likely to fit into a contact period than the complete message.

One example are opportunistic networks formed between mobile nodes (e.g., smartphones of mobile users). Contacts may often be short (e.g., when people walk past each other), thus limiting the amount of data that can be transferred. Fragmentation effectively increases the connectivity in such networks by allowing also shorter contacts to be exploited [3].

Two types of DTN message fragmentation are defined in [2]: *pro-active*, i.e., defining the fragment prior to transmission, and *reactive*. The latter means determining the fragment size based upon what got transmitted after a link went down so that no decision on the fragment size would need to be taken prior to transmission. However, this has two deficiencies: 1) Since duplicates detection is difficult with arbitrarily-sized fragments, reactive fragmentation may

lead to reduced performance [3]. 2) Cryptographic message validation, e.g., using message authentication codes (MACs) will not work with arbitrary-sized fragments since MACs would need to be available for every conceivable fragment size [4]. Both suggest introducing *quantization* of messages in evenly sized fragments of which as many as possible are sent together, which has been shown to be beneficial [3]. But this size has to be determined *proactively*.

Recent work [5]–[8] has investigated fragmented messages transmission over a *single* disrupted link, modeling packet or file transmission over a wireless link as well as single-hop forwarding of DTN messages. Typical DTN scenarios will, however, usually require message delivery over multiple links, which has not received attention so far.

In this paper, we address the case of message fragmentation over a chain of disrupted links. This case occurs, e.g., in a static multi-hop wireless network, where link disruptions can be due to transient channel effects. We are aware that, for opportunistic communication between devices of mobile users, message fragments will be sent by the routing protocol along different paths—i.e., multiple, possibly partly overlapping chains—towards a destination node [3]. But these multi-path scenarios cannot be unpacked without understanding the multi-link ones first.

We define a basic model for message transmission over  $n$  links in section II. The disruptions of communication links in the chain are characterized by i.i.d. ON/OFF periods. In section III we first identify the natural lower and upper bounds on the mean transmission time over  $n$  links. Then we derive a generic approximation formula for the mean transmission time. Estimates of the queue sizes in intermediate nodes are needed to compute this formula. In section IV we show how to compute these estimates in the cases of uniform and exponential distributions for disruptions. Using these results we can estimate the mean transmission times of fragmented messages in those cases.

To confirm our analysis we have computed the relative error between mean transmission times estimated with our formulas, and the actual transmission times in a simulated environment, where messages are transmitted according to our model over five disrupted links. From those experiments we conclude that our approximation is suitable for large message sizes, that are at least a few times bigger than what

can be typically transmitted within a single contact time; and the (relative) accuracy of our estimates increases with the message size.

## II. SYSTEM MODEL

The model used to obtain the analytical results is as follows. Network node A sends messages over a chain of  $n$  communication links to node B. Nodes are numbered  $0, 1, \dots, n$ ; node 0 is the sender A and node  $n$  is the receiver B.<sup>1</sup> The links change their state between ON and OFF independently from each other in a random manner. This arrangement is illustrated in Figure 1.

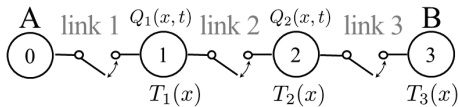


Figure 1. Schematic illustration of a chain of three disrupted links’ chain between the sender A (node 0) and the receiver B (node 3).  $T_1(x)$ ,  $T_2(x)$ , and  $T_3(x)$  are the mean transmission times of a message having size  $x$  over one, two, and three links, respectively.  $Q_1(x,t)$  and  $Q_2(x,t)$  model the average amount of data queued in the intermediate nodes 1 and 2 at time  $t$ .

The link speed during the ON state is constant (and the same) for all links.<sup>2</sup> We divide all message sizes by the (constant) link speed, measuring message sizes in seconds. The message size is denoted with  $x$ . The electromagnetic signal’s propagation times and the time it takes to acknowledge transmission over one link are neglected (zero) in our model.

The sending node A can choose to transmit the message in a single unit, thus requiring sufficiently long contact durations for the whole message to fit. Or A may split the message into blocks of size  $f$ , thus allowing transmission of message fragments consisting of one or more such (equal sized) blocks during shorter contacts.  $f$  may vary between messages, and we call it “message fragmentation unit”. Typical values of  $f$  could be  $10^{-3}$ ,  $10^{-2}$ , 1, all measured in seconds.

We assume, for simplicity, that message size  $x$  is an integral multiple of fragmentation unit  $f$ .

If the  $k$ th link in the chain is disrupted (fails) during transmission, node  $k - 1$  will attempt to retransmit the remaining message fragmentation units during the next ON epoch.

<sup>1</sup>In DTNs, paths of successfully delivered messages are often short because the network diameter is naturally constrained (as, e.g., in deep space networks), or messages do not travel very far in terms of distance and hops (as in mobile opportunistic networks). Therefore, we consider small values of  $n$ , say, less than 10, to be more interesting than large ones. But we do not exclude larger numbers of links in what follows.

<sup>2</sup>Please note that a sequence of ON/OFF epochs having different average link speeds during ON epochs, can be transformed into a sequence having constant link speed, that still retains same durations of OFF-ON epoch pairs as the original sequence. The details of this transformation are described in [8].

While the remainder of the message fragmentation units is still being transmitted over the first link, its head part consisting of fragmentation units already received in the first intermediate node, may be transmitted in parallel over the subsequent links. The same holds for all other links, but note that data bits inside the same fragmentation unit can be transmitted only over one link at a time; a node starts the transmission of a block, only if it has received all of that block.

We assume that the message transmission’s starting time  $t = 0$  is a random point in the sequence of ON/OFF periods. We denote time by  $t$ , and by  $t_k(x)$  the moments at which the transmission of a message with size  $x$  over  $k$ th link completes. We also agree that  $t_0(x) = 0$ .

The mean of  $t_k(x)$  is denoted with  $T_k(x)$ . When talking about the mean transmission time over a single link, we often omit the subscript; and write  $T(x)$  rather than  $T_1(x)$ .

The distributions of the link state durations (ON and OFF) are (i) the same in all links, and (ii) do not depend on time. Relaxing these assumptions to generalize our model is left for future study.

We denote with  $q_k(x,t)$  the amount of data queued in node  $k$  at time  $t$  during transmission of message size  $x$ . The mean of that random process at time  $t$  is  $Q_k(x,t)$ . The limit of  $q_k(x,t)$  when  $x \rightarrow \infty$  and  $k \geq 1$  describes the saturated input case, and is denoted by  $q_k(t)$ . It can be thought of as the  $k$ th queue size at time  $t$ , when the message size  $x$  is very large—much larger than any message we may wish to send over a chain of links. The mean of that random process at time  $t$  is  $Q_k(t)$ . From the system model it is clear that  $Q_k(x,t) = Q_k(t)$  whenever  $x > t$  because not more than  $t$  first seconds of the message may have been transmitted over any link at the time moment  $t$ . (This holds even in the limiting case where all links are in ON-state all the time.)

## III. ESTIMATING $T_n(x)$

Consider the mean transmission time  $T_n$  over  $n$  links. It follows from our system model that the mean transmission time of a message over  $k$  links is less than or equal to the sum of the mean transmission time  $T_k(x)$  of all message parts over the first  $k - 1$  links, and the mean transmission time  $T(x)$  (again, of all message parts) over the last,  $k$ th, link.

$$T_k(x) \leq T_{k-1}(x) + T(x), \quad (1)$$

where  $k = 1, 2, \dots, n$ .

The equality in the above equation holds only if the transmission over the  $k$ th link cannot start until all of the message parts have been received in the  $(k - 1)$ th node. This limiting case appears, for example, in transmission of a totally unfragmented message—then this kind of dependency exists between all adjacent links, and  $T_n(x)$  is the sum of  $T(x)$  values over individual links.

This limiting case gives us the natural upper bound on the mean transmission time  $T_n(x)$ .

The trivial lower bound on  $T_n(x)$  is given by another limiting case, where the  $n$  links are fully synchronous, i.e. they go ON and OFF simultaneously. Here the mean transmission time over  $n$  synchronous links equals that over a single link:<sup>3</sup>

$$T_n(x) = T(x). \quad (2)$$

Combining upper and lower bounds we have:

$$T(x) \leq T_n(x) \leq nT(x). \quad (3)$$

The upper bound in Eq. (3) implies that the growth of the mean transmission time with the number of links  $n$  is sub-linear.

In the rest of this section we derive an approximate formula for the mean transmission time in other cases (i.e. ‘in between’ those bounds).

Transmission time  $t_k(x)$  over  $k$  links, is the time  $t_{k-1}(x)$  it takes to transmit the whole message over the first  $k-1$  links, plus the time it takes to transmit the remaining data over the last,  $k$ th, link. The size of the remaining data in the  $(k-1)$ th node at the time  $t_{k-1}(x)$  is  $q_{k-1}(x, t_{k-1}(x))$ , and therefore:

$$t_k(x) = t_{k-1}(x) + t(q_{k-1}(x, t_{k-1}(x))), \quad k = 2, 3, \dots, n. \quad (4)$$

It follows that the mean transmission time over  $n$  links  $T_n(x)$  is the sum of  $T_{n-1}(x)$  and the mean of the last term:

$$T_n(x) = T_{n-1}(x) + E[t(q_{n-1}(x, t_{n-1}(x)))].$$

In our approximation we replace the last term with  $T(Q_{n-1}(T_{n-1}(x)))$ . (We replace all variables with the respective means, and replace the mean queue size  $Q_{n-1}(x, T_{n-1}(x))$  with the mean queue size  $Q_{n-1}(T_{n-1}(x))$  in the case of saturated input.) The result is a recursive formula that, together with the combined bounds of Eq. (3), characterizes  $T_n(x)$ :

$$T_n(x) \approx T_{n-1}(x) + T(Q_{n-1}(T_{n-1}(x))). \quad (5)$$

Please note that variants of Eq. (1-5) hold also in the case when the ON-OFF periods’ statistics are not the same in all links.

We now proceed to the question of computing Eq. (5). To compute this recursion we need to know how to estimate (i) the mean transmission times over a single link  $T(x)$  (for any message size  $x$ ), and (ii) the mean queue size  $Q_k(t)$  in intermediate nodes.

We already know how to do (i): The transmission time over a single link  $T(x)$  can be estimated using the results of our previous paper [7].

For the derivations below we restrict ourselves to the cases where the ON and OFF epochs are either uniformly

or exponentially distributed with the mean of one second in both cases. For the limiting case where the fragmentation unit  $f$  is the smallest possible, it is clear that we get a good estimate by  $T(x) = 2x$  because the link is in the ON state half of the time (in average) and it is possible to utilize the whole of each ON epoch for transmitting message fragments.

For the cases where the fragmentation unit  $f$  is larger, only part of ON epochs can be utilized for transmission, and consequently  $T(x)$  becomes longer. For the uniform distribution we obtain a relatively good approximation by simply including in the calculation only those ON epochs that are long enough for accommodating at least one fragmentation unit  $f$ . For example, in the case of  $f = 0.1$ s the proportion of those epochs is  $(2 - 0.1)/2 = 19/20$ , and  $T(x) = 2x/(19/20) = (40/19)x$ ; for  $f = 1$ s we get  $T(x) = 4x$ .

This kind of straight-forward approximation does not work as well for the exponential distribution, but slightly more complex reasoning (see the derivation of Eq. (V.17) in [7]) provides the following approximations for the same example fragment sizes: for  $f = 0.1$ s we have  $T(x) = 20x(e^{0.1} - 1)$  and for  $f = 1$ s we have  $T(x) = 2x(e - 1)$ .

Note that in all of these cases,  $T(x)$  is a linear function of  $x$ . This is asymptotically true also for any distribution of ON and OFF epochs with finite mean and variance (see Eq. (V.12) in [7]).

We will go into the question (ii) of estimating  $Q_k(t)$  in the next section.

#### IV. ESTIMATION OF MEAN QUEUE SIZES $Q_k(t)$

In this section we give estimates of queue sizes  $Q_k(t)$  in intermediate nodes in the cases of exponential and uniform distributions for disruptions. We begin by summarizing the results in §IV-A, then outline the estimation method in §IV-B, and end with the derivation details in §IV-C.

##### A. Summary of the results

Our approximation for  $Q_k(t)$  builds on the following estimate for the queue in the first intermediate node:

$$Q_1(t) \approx c\sqrt{t}, \quad (6)$$

where the constant  $c$  depends on distribution of ON and OFF epochs. We will show how to derive  $c$  for various distributions and give also explicit formulas for  $c$  in cases where ON and OFF epochs are distributed exponentially or uniformly.

We further have

$$Q_k(t) \approx a_k Q_1(t), \quad k > 1,$$

with the constants  $a_k$  tabulated below for  $k$  up to 10:

$k$	2	3	4	5	6	7	8	9	10
$a_k$	.63	.50	.42	.36	.33	.29	.27	.25	.23

Our approximation model is built in such way that these constants depend only on the means of the ON and OFF epochs distributions.

<sup>3</sup>Note that the lower bound could be tightened, because our system model assumes that fragmentation units cannot be forwarded by an intermediate node before they are entirely received. This causes an additional systematic per-hop delay that is a function of the chosen fragmentation unit size  $f$ .

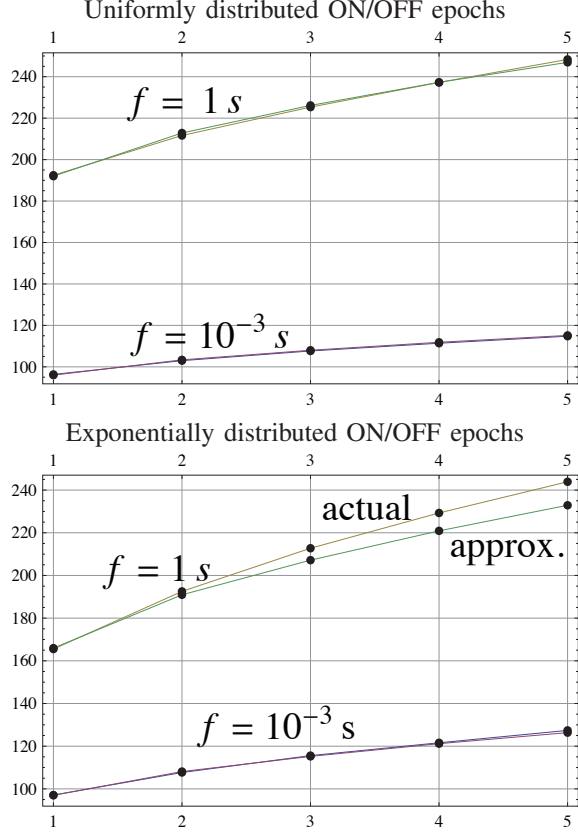


Figure 2. Plots of the actual and the approximated values of  $T_k(48s)$  for five links and two fragmentation unit sizes  $f: 10^{-3}s$ , and  $1s$ . The mean duration of ON and OFF epochs in all links is  $1s$ . The  $x$ -axis is the link number  $k = 1, 2, 3, 4$ , and  $5$ . The  $y$ -axis is the mean transmission time in seconds.

Combining these approximation formulas for the queue sizes with the approximations for  $T(x)$  from the previous section, we get the following formulas for our example distributions:

$$T_k(x) \approx T_{k-1}(x) + b_{k-1} \sqrt{T_{k-1}(x)}, \quad k > 1.$$

Here the constants  $b_k$  are as follows:

For the exponential distribution with a mean of  $1s$  we have

$$b_k = \begin{cases} 2a_k/\sqrt{\pi}, & f = 10^{-3} \text{ (smallest unit);} \\ 20(e^{0.1} - 1)a_k/\sqrt{\pi}, & f = 10^{-2}; \\ 2(e - 1)a_k/\sqrt{\pi}, & f = 1. \end{cases}$$

and for the uniform distribution (again, with a mean of  $1s$ ) we have

$$b_k = \begin{cases} (2/3)2a_k/\sqrt{\pi}, & f = 10^{-3}; \\ (2/3)(40/19)a_k/\sqrt{\pi}, & f = 10^{-2}; \\ (2/3)4a_k/\sqrt{\pi}, & f = 1. \end{cases}$$

Table I  
BIGGEST (IN ABSOLUTE VALUE) RELATIVE ERROR, IN %, BETWEEN THE ACTUAL AND THE ESTIMATED MEAN TRANSMISSION TIMES.

		$x[s]$	2	4	8	12	48
Exp.	$f [s]$	$10^{-3}$	2.4	0.9	-0.4	-0.8	-1.3
		$10^{-1}$	-1.8	-1.7	-1.9	-1.6	-1.6
		1	-15.2	-12.5	-10.4	-8.7	-4.5
Uni.	$f [s]$	$10^{-3}$	17.4	-1.0	0.8	1.0	-0.7
		$10^{-1}$	13.2	-3.2	-1.1	0.9	0.3
		1	12.9	4.9	3.5	2.9	0.6

To confirm our analysis we have written a custom simulator in C; it implements transmission of a message over a chain of  $n$  links according to our model.

We summarize next the simulation results in the 5-link cases, for both exponential and uniform distributions of ON/OFF epochs, and three fragmentation unit sizes  $f = 10^{-3}s$ ,  $10^{-1}s$ , and  $f = 1s$ . The mean value of the ON (or OFF) epochs is  $1s$ , in all cases; the message size  $x$  varies between  $2s$  and  $48s$ .

Each simulation run consisted of 100 single message transmissions. The start of the first message's transmission in each run coincides with the beginning of an ON epoch. Transmission of subsequent messages begins as soon as the previous message has been delivered to the destination. For each set of parameters we carry out 20 simulation runs with different random seeds and report on the mean results.

The results for  $x = 48s$  and comparison to the estimates given earlier in this section are plotted in Fig. 2. The data points corresponding to  $f = 10^{-2}s$  are omitted from Fig. 2 for clarity. It can be seen that the mean transmission time is concave with respect to the number of links  $k$ . The approximation for  $x = 48s$  is very good, except in the case of exponential distribution and  $f = 1s$ . (But we can see from Table I that even in this case the absolute value of the error is less than 5 %.)

In Table I we give the biggest (in absolute value) relative error values for message sizes  $2, 4, 8, 12$  and  $48$  seconds. The relative errors were computed as the difference between the actual (simulated) value of  $T_k(x)$  and our approximation to  $T_k(x)$ , divided by the actual value of  $T_k(x)$ . We see that: (i) The maximum absolute value of the error between our approximation and the actual  $T_n(x)$  decreases as the message size grows larger. The (maximal) errors are in general bigger for the exponentially distributed disruptions, than for the uniformly distributed disruptions. (ii) The threshold of  $\pm 10\%$  in the relative error value occurs between  $x = 8s$  and  $x = 12s$  for the exponential case; and between  $x = 2s$  and  $4s$  in the uniform case.

In summary, our approximation is good for large message sizes that are a few times bigger than the mean time between disruptions. The reason for this limitation is that the message transmission must last several ON/OFF epochs for our random walk model (described below) to apply.



### B. Estimation method

Recall that  $Q_k(t)$  describes in the transient behavior of the system with saturated input. (The input is saturated because the sender always has more data to send.) In some sense the only steady state of such system is the trivial case of the initially empty system. For that reason, steady-state solutions of queueing networks are not directly applicable in our case.

We use the concept of a random walk in our estimations of  $Q_k(t)$ ; this is commonly used in queueing theory [9].

A queueing process can be coupled with that of a one-dimensional ‘walk’ in the following way: The walker takes only a single step of size  $\pm 1$  in each move. The amount of queued data is zero at the initial position of the walker. After the walk starts, the amount of queued data increases by a constant  $l$  if the walker makes a move to the right (i.e. when the position of the walker increases by 1); and it decreases, again by  $l$ , if the walker makes a move to the left (i.e. when the position of the walker decreases by 1). However, there is an additional important rule: the amount of the queued data is always non-negative; e.g., if the queue is empty and the walker makes a move to the left then the size of the queue remains as zero.

Let us denote with  $s_k$  the position of the walker relative to its starting point after  $k$  moves, and with  $m_n$  the minimum of  $s_k$  after  $n$  moves:

$$m_n = \min_{0 < k \leq n} (s_k).$$

It can be shown from the coupling above that the amount of queued data is  $l(s_n - m_n)$ . The mean queue size is, therefore,

$$l \cdot (E[s_n] - M_n), \quad (7)$$

where  $M_n$  denotes  $E[m_n]$ .

Also, the number of time points when the corresponding queue is empty equals to  $-m_n$ .

In our derivation below, we define a walk coupled with  $q_1(t)$  that makes  $n(t)$  moves up to time  $t$ , and compute an estimate of  $E[s_{n(t)}]$ , and  $M_{n(t)}$  for that walk. An approximation of  $Q_1(t)$  is obtained then from Eq. (7).

This procedure is repeated for each subsequent queue in the intermediate nodes, up to  $Q_{10}(t)$ . The procedure for subsequent queues  $Q_2(t)$ ,  $Q_3(t)$  etc. is more complex than for the first queue, because the walk is affected by the cases when the previous queue is empty. At these time points the succeeding queue cannot ever increase. We will go into details in the next section.

### C. Derivation details

Whenever there is an ON epoch on the first link, then at least a part of the message can also be transferred. In other words, during the message transmission over the first link the queue in the sending node A behaves in a simple way:  $q_0(x, t)$  is decreasing for all values of  $t$  that belong to an ON

epoch (of the first link) and  $q_0(x, t)$  is stable for all values of  $t$  that belong to an OFF epoch (of the first link).

Let us consider now  $q_1(t)$ , the length of the queue in the first intermediate node when we assume that the source node always has more data to transmit. In a certain sense, we have a relatively simple situation.

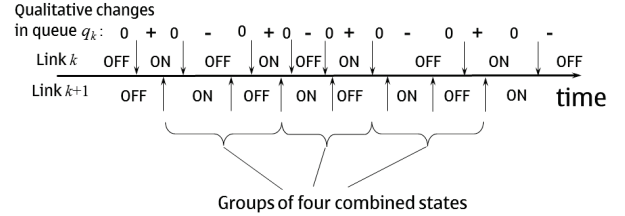


Figure 3. Combined states between two consecutive links

There are four possible cases for the behavior of  $q_1(t)$  depending on whether the first link/second link is ON/OFF. For values of  $t$  where the combined state of the first two links is either ON/ON, or OFF/OFF, the queue length  $q_1(t)$  remains stable. For values of  $t$  where we have ON/OFF, the queue becomes longer while, for points in time where we have OFF/ON, the queue becomes shorter (unless it is already empty).

Next we show how the concept of a combined state can be turned into a random walk model. Let us first exclude the possibility that two consecutive links would change their states in exactly the same moment in time. (It is clear that this assumption does not introduce any big error.)

A sequence of ON/OFF states in two consecutive links is illustrated in Fig. 3. We notice that every second combined state (time) interval keeps the queue size  $q_1(t)$  unchanged. From the other combined state intervals, typically every second (altogether every fourth) would increase the queue size while the rest (altogether also every fourth) would decrease the queue size. This motivates grouping four consecutive combined state intervals together. An expected length of such group is two seconds in case the mean of each epoch is one second.

Now the behavior of the random variable  $q_1(t)$  can be modeled by a random walk where we would take a random walk move every 2 seconds. The length of the random walk move depends on the length difference between the ‘increased queue’ time interval and the ‘decreased queue’ time interval. The model is based on the assumption that these differences for two consecutive groups (each consisting of four consecutive combined states) do not depend on each other too much.

The distribution of the above length difference depends on the distributions of ON and OFF patterns. On this aspect, uniform and exponential distributions behave differently. If we have two i.i.d. variables  $U$  and  $V$  whose mean is one second and the distribution is uniform (resp. exponential) then

(it is easy to calculate that) the absolute difference  $|U - V|$  has the mean of  $2/3s$  (resp.  $1s$ ). Although the combined state intervals do not have exactly uniform (resp. exponential) distributions, this however gives a pretty good approximation for the purposes of our model.

Summary of the model for  $q_1(t)$  is the following. We have a random walk model for the queue size  $q_1(t)$  in which one move is taken every  $2s$  to the left or to the right with equal probability;  $n(t)$  is approximately  $t/(2s)$ ; the length of the move is  $1s$  in the case of exponential distribution of ON/OFF epochs, and it is  $2/3s$  in the case of uniform distribution. We denote this length with  $l$ . As discussed in the previous section, an additional constraint is imposed by the natural fact that the queue size is never negative. Random walk theory provides now formulas for the mean queue size  $Q_1(t)$  (and also for frequencies of empty queue instances).

We return to these formulas in a moment, but first we note that the situation is slightly different for the next queue  $q_2(t)$ . Whenever the first queue is empty, i.e.  $q_1(t) = 0$ , then the queue size  $q_2(t)$  cannot increase. This implies a negative drift in the walk related to  $q_2(t)$ . The same is true for  $q_3(t)$  etc. Moreover, the negative drift implies smaller average queue sizes  $Q_2(t)$  (when compared to queue sizes  $Q_1(t)$ ).

Also, the queue in the second intermediate node is more often empty than the queue in the first intermediate node, which in turn implies even bigger negative drift for random walk related to  $q_3(t)$  etc.

As a conclusion of this reasoning, it seems that in the intermediate nodes  $Q_k(t) > Q_{k+1}(t)$ . (Please recall here that we defined  $Q_k(t)$  in such way that the queue in the source node is never empty.)

Now we turn our attention to the actual formulas for queue sizes  $Q_k(t)$ .

For the simple random walk the mean position of the walker relative to its starting point after  $n$  moves  $E[s_n]$  is zero, and the ratio between the expected minimum  $M_n$  and  $\sqrt{n}$  tends to the constant  $-\sqrt{2/\pi}$  as  $n$  grows.<sup>4</sup> Thus  $M_n \approx -\sqrt{2n/\pi}$ . (This estimate is rather good already for moderate values of  $n$ : e.g., at  $n = 50$  the error is less than 10%.) In our case  $n(t) \approx t/(2s)$ , and so  $M_{n(t)}$  is approximately  $-\sqrt{t/\pi}$ , where  $\pi$  has the dimension of seconds. Therefore, by Eq. (7)

$$Q_1(t) \approx l\sqrt{t/\pi}.$$

The constant  $c$  in Eq. (6) is  $l/\sqrt{\pi}$  and has the dimension  $s^{1/2}$ .

The value  $\sqrt{t/\pi}$  is also an estimate to the expectation for the number of time points when the queue in the first intermediate node  $q_1(t)$  is empty. As explained above, this fact gives us the means to estimate the negative drift that

<sup>4</sup>By symmetry considerations the  $-M_n$  equals to the expected maximum of simple random walk. The limit for the ratio between the expected maximum and  $\sqrt{n}$  is given in, e.g., [10] p. 235, Eq. (4.8.23).

affects the mean queue size in the second intermediate node  $Q_2(t)$ . The drift is referred to by  $-\delta\sqrt{t}$  in Fig. 4.

Unfortunately, the drift is a nonlinear function of  $t$ , and therefore  $q_2(t)$  cannot be modeled directly by a (proper) random walk. Instead, we model  $q_2(t)$  with a sum of a simple random walk (like the one in the model of  $q_1(t)$ ) and the nonlinear negative drift function  $-\sqrt{t/\pi}$ .

The expected end point  $E[s_{n(t)}]$  of this combined walk is  $-\sqrt{t/\pi}$ , and by the formula (7):

$$Q_2(t) \approx l \cdot \left( -\sqrt{t/\pi} - M_{n(t)} \right). \quad (8)$$

It is a little bit tricky to estimate  $M_{n(t)}$ , because it depends on the nonlinear drift component. On the one hand, the nonlinear negative drift naturally pushes the time point, say  $\gamma$ , where the minimum  $m_{n(t)}$  is reached closer to the end point  $t$ , compared to the case where  $\gamma$  is determined by the random walk component alone. It can be shown that without the drift component

$$\Gamma = E[\gamma] \approx t/2.$$

On the other hand, this kind of bias in the “selection” of minimum point  $\gamma$  restricts the amount that the random walk component contributes to  $M_{n(t)}$ . (If the minimum would depend only on the drift then, of course, we would have  $\gamma = t$ , and the contribution of the random walk component would be zero.) The component of the minimum contributed by the negative drift is marked as  $-\beta\sqrt{t}$  in the Fig. 4.

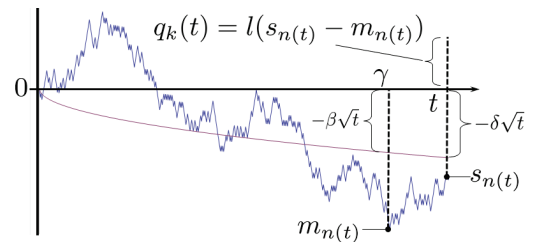


Figure 4. Illustration of the queue  $q_k(t)$ , and the related random walk with negative drift.

We try to approach these two aspects sequentially, starting by an estimate for  $\gamma$ . We use a simple triangular approximation for the probability density function of  $\gamma$  between 0 and  $t$ , assigning density of 0 for the time point  $\tau = 0$  and increasing the density function linearly towards  $2/t$  at time point  $\tau = t$ . Then we obtain  $\Gamma \approx (2/3)t$ .

Using this estimate for  $\Gamma$  we approach the deviation of  $M_{n(t)}$  from the negative drift function at point  $\Gamma$ . The closer  $\Gamma$  is to  $t$ , the smaller is the absolute value of the expected deviation.

We give a simple estimate for this deviation as well. A natural choice for an estimate is given by the value  $-\sqrt{2(1-\Gamma/t) \cdot (t/\pi)}$ : The deviation corresponding to  $\Gamma = t/2$  can be approximated by the expected minimum of

the random walk component,  $-\sqrt{t/\pi}$ ; and as  $\Gamma$  increases towards  $t$ , the absolute value of the approximated deviation decreases proportionally towards zero.

These two estimates can now be put together to find an estimate for the expected minimum  $M_{n(t)}$  of the combined walk, and subsequently also for  $Q_2(t)$ : At  $\Gamma = (2/3)t$  the drift, as well as the deviation, equal to  $-\sqrt{(2/3)(t/\pi)}$  each. Therefore,  $M_{n(t)} \approx -2\sqrt{(2/3)(t/\pi)}$ , or  $-1.63\sqrt{t/\pi}$ . By Eq. (8),

$$Q_2(t) \approx l(0.63\sqrt{t/\pi}).$$

For the next node and the queue size  $Q_3(t)$  we repeat similar calculations. The negative drift  $-\delta\sqrt{t}$  is now bigger (namely,  $-1.63\sqrt{t/\pi}$ ) and therefore  $\Gamma$  should be closer to  $t$ .

This is captured by fine-tuning the triangular approximation to the probability density of  $\gamma$  a little bit: we assign density of 0 to all points  $\tau \in [0, t]$  where  $\delta\sqrt{\tau} < \sqrt{t/\pi}$ . (At those points the magnitude of the negative drift  $\delta\sqrt{\tau}$  is still less than the magnitude of the random walk component's expected minimum  $\sqrt{t/\pi}$ .) After this adjustment, the probability density function of  $\gamma$  is 0 for  $\tau \leq \tau_0$ , where  $\tau_0 = (1 - 1/(\sqrt{\pi}\delta))^2 t$ , and it increases linearly from 0 at  $\tau_0$ , towards  $2/(t - \tau_0)$  at time point  $t$ .

Now we obtain  $\Gamma = 0.717t$ . The rest of the calculation is as described in the  $Q_2(t)$  case, resulting in  $M_{n(t)} \approx -2.14\sqrt{t/\pi}$ , and

$$Q_3(t) \approx l(0.50\sqrt{t/\pi}).$$

For the subsequent queue sizes we can repeat exactly the same calculations as were done for  $Q_3(t)$ . They lead to the end results summarized earlier in section IV-A.

## V. CONCLUSION AND FUTURE WORK

We have shown how to estimate mean transmission times of fragmented message over a chain of disrupted links, where the link disruptions are identically and independently distributed. We validated our approximation formulas by simulating the cases of uniform and exponential distributions for disruptions. From those simulations we conclude that our approximation is suitable for large message sizes, that are at least a few times bigger than what can be typically transmitted within a single contact time, and its accuracy increases with the message size.

An immediately obvious item for future work is estimating the mean transmission times for smaller messages. Other future work topics on the multi-link chain scenario include estimating the variance of the mean transmission time, verifying our approximation with additional distributions of contact and inter-contact times, and analyzing the case where statistics of those times for different links are not the same. Based upon those, our target is (as mentioned in the introduction) investigating fragmented message transmission over multiple paths in mobile opportunistic networks.

## VI. ACKNOWLEDGMENT

This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

## REFERENCES

- [1] K. Fall, "A delay-tolerant network architecture for challenged internets," in *SIGCOMM '03*. New York, NY, USA: ACM, 2003, pp. 27–34.
- [2] K. Scott and S. Burleigh, *Bundle protocol specification*, IETF RFC 5050, November 2007.
- [3] M. Pitkänen, A. Keränen, and J. Ott, "Message fragmentation in opportunistic DTNs," in *Proceedings of the Second WoWMoM Workshop on Autonomic and Opportunistic Communications (AOC)*, 2008.
- [4] C. Partridge, "Authentication for fragments," in *Proceedings of the Fourth Workshop on Hot Topics in Networks (HotNets-IV)*, College Park, Maryland, USA. ACM, November 14–15 2005.
- [5] P. R. Jelenković and J. Tan, "Dynamic packet fragmentation for wireless channels with failures," in *Proceedings of MobiHoc'08, Hong Kong, China*. ACM, May 27–30 2008, pp. 73–82.
- [6] J. Nair, M. Andreasson, L. Andrew, S. Low, and J. Doyle, "File fragmentation over an unreliable channel," in *Proceedings of INFOCOM 2010, San Diego, CA, USA*. IEEE, March 14–19 2010.
- [7] P. Ginzboorg, V. Niemi, and J. Ott, "Message fragmentation for disrupted links," in *Proceedings of European Wireless 2011, Vienna, Austria*. VDE VERLAG GMBH, April 27–29 2011, pp. 415–424.
- [8] —, "Message fragmentation algorithms for disrupted links," submitted for publication.
- [9] N. Prabhu, *Stochastic storage processes: queues, insurance risk, dams, and data communication*. Springer Verlag, 1998, vol. 15.
- [10] A. Rényi, *Foundations of Probability*. Dover, 1998.