

Optimising Replication versus Redundancy in Window-aware Opportunistic Routing

Gabriel Sandulescu
University of Luxembourg
gabriel.sandulescu@uni.lu

Simin Nadjm-Tehrani
Linköping University, Sweden
simin@ida.liu.se

Abstract—In this paper we address the replication versus redundancy optimisation problem for a class of intermittently-connected ad hoc networks with the following three characteristics: routing is performed by a store-carry-forward mechanism in an opportunistic manner, replication of custodian messages is used to increase the delivery ratio, and fragmentation combined with redundancy in the form of erasure coding is used to deal with large messages in short contact windows. We propose a novel mathematical model in which the merits of a high replication factor can be compared to a high level of redundancy in erasure coding. The purpose of the model is to identify the optimised level of replication and redundancy as a function of application-specific parameters like message size and time-to-live. Our model builds upon a component for message latency distribution that we have borrowed from earlier works (SprayAndWait) and that has been extended for erasure coding and constrained resources. The model is general and can be applied in different opportunistic settings using the following approach. First, we derive a set of equations that are demonstrated to identify the optimal replication and redundancy factor in a network with constrained resources. Then, the paper includes an extension that provides a tailored latency distribution based on history profiling, thus making it applicable to any protocol that uses replication/redundancy in a similar network. By theoretical analysis and simulations, we demonstrate that using erasure coding together with replication offers an efficient yet flexible tradeoff between resource cost and protocol performance.

Keywords-communication theory; delay-tolerant communications; opportunistic; erasure coding; performance

I. INTRODUCTION

Opportunistic routing in delay-tolerant networks (DTN) makes no assumptions about node contact schedules [1]. To increase the chances of successful delivery, some routing algorithms forward multiple copies of each message to several custodians. However, this also consumes resources (bandwidth and implicitly energy, as well as storage space at custodians) in proportion to the number of copies forwarded. Some of the more well-known examples are the Epidemic protocol [2] and the SprayAndWait protocol [3]. The latter algorithm mitigates the overhead problem of epidemic schemes by limiting the number of copies maximally found in the network, also referred to as the *replication factor*. The original SprayAndWait scheme has 2 phases. In the first phase, copies are forwarded to a number of nodes

encountered, according to the replication factor L . In the second phase, nodes carry the copies until direct encounter with the destination in order to deliver the message.

An alternative method of improving opportunistic routing performance is to erasure-code a message, and distribute the generated code-blocks potentially over multiple paths. Sending smaller data units in an opportunistic network better accommodates cases when bandwidth, contact time and buffer space are limited. Instead of sending a copy of the message over one relay, only a fraction of the code-blocks is sent at each forwarding opportunity. A *redundancy factor* k can then be chosen so that a message of size s is split into n , s/m -sized blocks ($k = n/m, k \geq 1$). Encoding is based on Reed-Solomon or Tornado codes which allow reconstruction of the original message from any m different fragments arriving at the destination.

In this paper we consider the following scheme: the initial message is erasure-coded at source by applying a redundancy factor k , and then every fragment is delivered independently by a protocol using controlled replication with a factor L . We study the impact on performance of the combined scheme as an instance of an opportunistic routing mechanism that employs both replication and redundancy. In particular, we build an approximate mathematical model that describes the probability of successful delivery of a message as a function of various parameters, such as time-to-live, number of nodes, fragment size and degree of replication and redundancy, respectively. We propose the use of this model in order to optimise performance for a given time-to-live parameter in terms of minimising the overheads associated with replication and redundancy.

The contributions of this paper are as follows: (1) a mathematical model for optimising combined replication and redundancy parameters on a per message basis, or on a network basis for uniform networks, and (2) a proposal to adapt the parameters over time, depending on actual behaviour, and recorded latencies.

We propose a model that, based on fragment size, as derived from network mobility assumptions, computes the probability of delivering a complete message before a given time-to-live. Thus, it provides a scheme for determining replication and redundancy pairs (L, k) that can then be

optimised according to a cost function. The model includes a probability distribution component for message latency. This component is initially derived from earlier equations for expected latency as proposed by Spyropoulos et al. [4], [5]. We adapt those equations by (1) integrating erasure coding into the model, and (2) adapting it to resource-constrained networks. The latter revises the original assumption that contact time and buffer space is no limitation in forwarding, thus making the model more realistic. We use numerical studies as well as network simulations to validate our model and to demonstrate the practical benefits of using this optimisation.

Next we propose a framework for optimising a generic protocol using controlled replication and erasure coding. This works in two phases, the initiation phase and the adaptation phase. In the initiation phase the basic model is used to obtain the presumed optimised redundancy and replication factors, given a desired probability of message reconstruction at destination within a given time-to-live. Following this phase, actual data on average latency is stored for various source-destination pairs. In the adaptation phase, the recorded distribution for (real) average latency replaces the initial approximation, thereby producing new settings for the replication and redundancy factors that are closer to the network, mobility model and protocol characteristics. This makes the method generically applicable to any opportunistic protocol that uses controlled replication and erasure coding, e.g. ORWAR routing protocol as introduced in [6] and extended with erasure coding in [15]. While the papers mentioned above described protocol performance using simulation runs, this paper adds a new element by presenting a mathematical framework for optimisation.

Demonstrating the use of the optimisation model, the paper includes analyses giving the following insights: (1) an appropriate and correlated level of replication and redundancy makes a visible difference in terms of network level costs, given the same latency targets, (2) network overload, materialised as diminished available bandwidth or shortened buffer at custodians, affects cost in a significant way, for instance best cost is found at a particular value for overload, which is neither at the bottom nor at the top of the range, and (3) erasure coding a message in very small chunks - much lower values than those required by a typical contact window - negatively affects cost by ineffectively multiplying delivery paths. Initial simulations suggest that this framework is a practical tool for optimising routing parameters in a challenged network.

The structure of the paper is as follows. In Section I we provide a general introduction to the paper, while in Section II we describe the background and make an overview of related work. In Section III, we state our assumptions and formulate a method to estimate an appropriate fragment size for opportunistic mobile routing. In Section IV we use that fragment size as an input to a combined

replication/redundancy model based on an extension of the equations for SprayAndWait and Oracle based delivery. We have devoted Section V to the study of our combined model by simulations. These are compared to numerical results, and shed a light on the respective merits of replication versus redundancy when costs are minimised.

In Section VI we have also introduced an adaptive element that can be added to any algorithm that uses such an optimisation element in order to improve its grounding in realistic settings. Finally, Section VII concludes the paper.

II. BACKGROUND AND RELATED WORK

There are numerous examples of protocols using message replication as a mechanism to increase delivery ratio. Replication may be seen as flooding, such as in different variations of the Epidemic scheme [2], or as controlled replication, such as in SprayAndWait [3], ORWAR [6]. The latter two are multi-copy routing schemes where a fixed number of copies is distributed over the network. Under some simplifying assumptions, such as basic mobility patterns, a large number of nodes and small message sizes, some of these protocols can be studied analytically. Zhang et al. [7] obtain a rich set of closed form formulas on delivery delay and on number of copies sent under the following extended epidemic schemes: k -hop forwarding, probabilistic forwarding, and limited-time forwarding. One of these extended schemes, probabilistic forwarding, was first proposed by Lindgren et al. [8] and then further developed by Haas et al. [9]. Another approach proposed by Lindgren et al. [8] and continued by Spyropoulos et al. in SprayAndFocus [10] was that encounter history can determine future delivery probability. In section VI, we leverage on this by reusing history to aid optimisation. More specifically, delivery history is used to get delivery time distribution that is further used in cost optimisation.

The ability to fragment bundles, either prior to transmission (proactive fragmentation) or while in transit (reactive fragmentation) has been introduced early in the DTN design [11]. However, the vast majority of routing protocols consider bundles as indivisible, making routing decisions simpler. Conversely, Jain et al. formulate the problem of optimising the probability of successful message delivery by splitting and erasure coding [12]. Moreover, nodes may send out packets with linear combinations of previously received information leading to network coding [13]. In other cases, distributed caching is studied as a particular case of applied redundancy [14].

Our earlier work [15] also introduces erasure coding in a simulated environment but does not include any analytical models. In our current work, we use replication and erasure coding as components in an analytical model that can be used for protocol parameter optimisation.

The relation between limited contact windows and the size of the data sent was studied by simulations in a city mobility context [15] or by real-life implementations [16]. Sending

smaller data units can be considered as a simple solution for diminishing partial transmissions. However, when the desired level of data exchange exceeds the capacity of a contact window, some messages should be preferred to others. In this case, message differentiation using a utility function may be useful, as shown in [6]. Pitkänen et al. [17] discuss the choice between proactive fragmentation and reactive fragmentation and show that both can accommodate limited contact windows. In this paper we use proactive fragmentation as a result of erasure coding and we argue that parameters such as replication factor, redundancy factor and data unit size can be initially optimised based on an analytical model, later adapted to real mobility conditions. The relation between node speed, radio range and the mobility pattern was also studied by Groenvelt et al. [18] using a rigorous analytical framework, but it was limited to three mobility patterns and, more importantly, it assumed infinite bandwidth during contacts.

In another theoretical paper, Altman et al. [19] proposed a tradeoff between transmission delay and energy consumption by letting a replication protocol deliberately bypass some contact opportunities - in order to conserve energy - and still deliver within the required delay. However, message coding was not considered, nor any other solution that takes into account limited contact windows or real message sizes. In our case, since messages are split by erasure coding, they can also take advantage of short contact opportunities, particularly in very dynamic networks. Unlike erasure coding, network coding was studied by Lin et al. [20] as an add-on to replication. However, network coding and erasure coding are essentially different. One particular difference is that network coding requires all intermediary nodes to be aware of the protocols involved, with important computational requirements, that are rather burdensome and have a heavy overhead. In our case, we propose a solution relying on source-based erasure coding, so that computation is only involved at source and, to a lesser extent, at destination. Moreover, as the routing mechanism between source and destination is not affected, we expect that the optimisation mechanism remains suitable for a wide range of routing protocols using replication.

III. MODELLING FRAGMENT SIZE

Given a certain mobility pattern, a node may be within communication range of a forwarding node or its destination for a short time only. A *contact window* between two nodes is defined as the time period during which the two nodes are in communication range.

As shown in our earlier work [15], when nodes are mobile devices, it is worthwhile for them to exchange vectorial speed, radio range and direction of movement when they meet. By doing this, contact windows can be estimated using locally available knowledge, so that estimated maximum

transferable message size over the current contact window is not exceeded.

A. The wireless model

In our analytical approach we consider a simple model based on the common assumptions made in simulations of wireless networks, as presented by Kotz et al [21]. Although nodes may be located anywhere in 3D space, we assume that our nodes are in a 2D space. Disregarding atmospheric conditions, node placement, orientation, and battery status, we consider all nodes as having equal radio range r . Even if real world radio ranges are neither regular, nor circular or contiguous, we consider them as perfect circles. We consider transmission to be perfectly symmetrical. We specifically ignore obstacles that cause signal obstruction, diffusion or scattering. Our model is built for a sparse network, as it is often the case in Delay and Disruption Tolerant Networks. Therefore we limit our model to contacts between node pairs only.

B. Contact window

Given 2 nodes advancing at a vectorial speed of \vec{v}_1 and \vec{v}_2 respectively, having the radio range r , we can calculate the contact window time t_{cw} as being:

$$t_{cw} = \frac{2 * r * \cos \alpha}{|\vec{v}|} \quad (1)$$

where $\vec{v} = \vec{v}_1 - \vec{v}_2$ and α is the angle between \vec{v} and the line defined by the 2 nodes at contact time, as depicted in Figure 1.

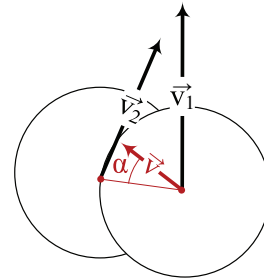


Figure 1. Contact window calculation

Assuming a data transmission rate b , we can calculate the maximum exchangeable message size $s_{max} = t_{cw} * b$ over the current contact window.

One approach to optimising network resources is to send only messages that fit into a contact window at each meeting (thus reducing partial transmissions). This approach has, for example, been adopted in our earlier work [15] that deals with window-aware routing. The mathematical model in this paper will build on an alternative approach where an estimation of maximum network-wide fragment size is used as an upper bound for data exchange unit size.

C. Constant fragment size

With smaller data units, computing the actual contact window at each meeting provides less incentive.

In equation (1), α and \vec{v} are affected by the mobility model, therefore they can be evaluated under a specific scenario. Let us analyse, for example, the case of a city scenario where nodes are carried by pedestrians, cars and trams. We argue in [15] that most of the contacts are established between nodes having rectilinear and parallel trajectories, defined by roads. Moreover, the speed interval is also known, so that the worst-case scenario can be determined. We conclude that fragment size f may be estimated as being:

$$f = 2 * r * b * c_m \quad (2)$$

where c_m is a mobility model constant which can be experimentally estimated. In our simulation context [15], c_m was about 0.1 [s/m].

Adopting a size for the exchanged data units that is below but close to f significantly improves energy consumption by diminishing partial transmissions [6]. This is true even though nodes cannot send all messages that are candidates for forwarding at a given meeting. Although nodes do come into contact, the contact window may be too short or buffer space at custodians may be inadequate. Consequently, only a subset of the initially scheduled messages will be successfully transmitted. Let us denote with β the probability that a data unit is successfully forwarded over a given contact and assume that this probability is uniformly distributed over all messages and all contacts in a given network. We can assume that this probability β only affects the forwarding process and not the final delivery process. In fact, because the final delivery is prioritised¹ over the forwarding process, we assume that all final deliveries are successful whilst forwards to custodians are successful only for a β -equivalent subset of those messages, at each contact.

Equation (2) allows us to calculate the number of fragments m required to arrive at destination for a specific message size s , whilst exchanged data unit size remains constant over the network. As we show in the next section, we need this value (m) in order to calculate the required number of fragments (n) according to the redundancy factor ($k = n/m$).

IV. THE MATHEMATICAL MODEL

In this section we propose a mathematical model that evaluates the probability of successful message delivery before a given time, as well as parameters that describe degrees of replication and redundancy. We will use the

¹Meeting the destination is a precious opportunity that no routing algorithm can afford to miss, typically treated as direct delivery. Moreover, in these cases, no buffer space is needed, as the application layer will just consume the data once the destination is reached.

model to calculate when it is more efficient to use one technique or the other (replication or redundancy) for a given (fixed) probability of message delivery while minimising delivery costs.

A. Network, mobility and routing assumptions

We consider a set of M nodes moving in a closed area A , and different source-destination pairs. The mobile nodes are equipped with a wireless device with a fixed transmission range r that is small compared to area A . We assume that all nodes move according to some mobility models, whose meeting times are almost exponentially distributed. We consider that at each meeting the nodes can exchange a limited number of messages according to the contact window, message size and available buffer space at custodian, this being captured by the forwarding probability β - assumed to be uniform over the network.

We assume that the message is erasure coded at source, then routed to destination over independent paths, which is a reasonable assumption for a big enough M . Moreover, considering a small β , only small subsets of the message list will be forwarded at each contact, thus multiplying message paths between source and destination.

As for erasure coding at source and reconstruction at destination, we consider that these processes occur with negligible delay and with negligible effect on system resources (primarily bandwidth and transmission energy).

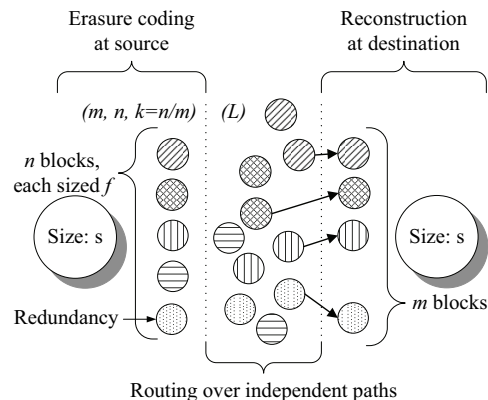


Figure 2. Erasure coding and routing using replication

In Figure 2 we show an initial message being erasure coded, then routed to destination over independent paths. Our example depicts $n = 5$, $m = 4$, $k = 1.25$ and $L = 3$. We have no further assumptions on the routing protocol except that it should rely on controlled replication and allow erasure coding.

B. Delivery time distribution without erasure coding

In order to find the probability of delivering a message before a certain time, we need to find the distribution of message delivery in a network that uses replication. Earlier

work has studied delivery metrics using binary SprayAndWait with no erasure coding in a uniform network, and this is an element we will reuse here.

Considering that meeting time between nodes follows a Poisson distribution, Groenveld et al. [18] show that process intensity λ can be calculated for 3 mobility models (random walk, random waypoint and random direction). We henceforth use the mean pairwise meeting time $\tau = 1/\lambda$ as a parameter in the model. Table I provides some other variables that will be used in our model.

Variable	Definition
τ	mean pairwise meeting time
M	total number of nodes in the network
T_o	expected latency under Oracle routing
T_{sw}	expected latency under binary SprayAndWait routing
$T_{sw}(n)$	expected latency under binary SprayAndWait routing and erasure coding, with n fragments
β	forwarding probability of a message during a contact
L	maximum number of copies in the network (protocol parameter for controlled replication)
T	message time-to-live
p_T	probability that one message is delivered before time T
$p_T(n)$	probability that n fragments are delivered before time T
s	message size
f	fragment size
m	number of fragments (excluding redundancy, calculated as $\lceil s/f \rceil$)
n	total number of fragments per message with redundancy
k	redundancy factor ($k = n/m$)

Table I
SUMMARY OF VARIABLES USED

Given a certain network model, we can estimate performance in the presence of the optimal algorithm, known as Oracle routing. Spyropoulos et al. [5] show that expected delivery time can be approximated as follows:

$$T_o = \frac{\sum_{i=1}^{M-1} (1/i)}{(M-1)} * \tau \quad (3)$$

T_o acts as a lower bound on expected delay for every other routing algorithm that would have been deployed in this given network. For a network deploying replication, in another work [4], Spyropoulos et al. detail the upper bound for the expected delay when binary SprayAndWait routing protocol is used:

$$T_{sw} = \underbrace{\sum_{i=1}^{L-1} \left(\frac{\tau}{(M-i)} \right)}_{\text{forwarding (spray)}} + \underbrace{\left(\frac{(M-L)\tau}{(M-1)L} \right)}_{\text{final delivery (wait)}} \quad (4)$$

In equation (4), we highlight the two components of the SprayAndWait delay: the forwarding or spraying phase, and the final delivery, also known as the waiting phase.

We consider therefore that, for a given replication factor (L) and network parameters (M and τ), a message routed

by SprayAndWait will have an expected latency bounded by the following 2 values: T_o and T_{sw} . The next question is what the shape of the distribution is for delivery delays in between these two bounds.

We have conducted extensive simulations to find the distribution of arrival latency for messages arriving between T_o and T_{sw} .

Figure 3 shows the distribution of delivery times gathered in the simulation environment for 2 different replication rates L when using classical SprayAndWait routing. On the x-axis we show 10-minute time intervals, while on the y-axis we show the corresponding proportion of the total messages delivered within this time interval. Results are simulated using ONE [22] in a 100-node network² moving in a random waypoint pattern. Other simulations (not shown here) confirm that the shape is very similar also when the nodes follow a random walker and map-based mobility model, as well as when different replication rates L are used.

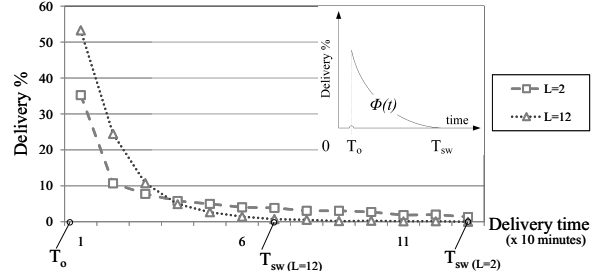


Figure 3. Latency distribution - simulation results

If we can approximate the above behaviour with a mathematical distribution, then it can be used to calculate the probability of a message to be delivered before its time-to-live T expires. In most cases, message latency is below T_{sw} and a meaningful approximation should cover a large area under the curve by time T_{sw} .

We have chosen a simple logarithmic approximation, as shown in Figure 3. Let us denote this approximation by $\Phi(t) = a * \ln(t) + b$ with $\Phi(T_{sw}) = 0$ and $\Phi(T_o) = 1$. By integrating $\Phi(t)$ between T_o and T we can calculate the expected probability p_T that the message will be successfully delivered before time T :

$$p_T = \begin{cases} 0 & \text{if } T < T_o \\ \frac{\ln(\frac{T}{T_o}) - (T - T_o)\ln(\frac{e}{T_{sw}})}{\ln(\frac{T_{sw}}{T_o}) - (T_{sw} - T_o)\ln(\frac{e}{T_{sw}})} & \text{if } T_o < T < T_{sw} \\ 1 & \text{if } T > T_{sw} \end{cases} \quad (5)$$

²Section V provides details of the simulation setup.

Equation (5) shows that, under our assumptions, the probability p_T of a message reaching its destination before time T can be approximated when we know M , L and τ . For this equation, we assume simple message delivery without erasure coding, as well as successful forwarding at every physical encounter. The next section extends the model to deal with these two factors.

C. Delivery time distribution with erasure coding and constrained resources

In this subsection we augment the previous equations to deal with cases where (1) erasure coding is used, and (2) message forwarding is not automatically successful at every physical encounter. We can calculate the upper bound for expected latency for n fragments as being:

$$T_{sw}(n) = \sum_{i=1}^{L-1} \left(\frac{\tau}{\beta * (M-i)} \right) + \frac{(M-L)\tau}{(M-1)L} + \sum_{i=1}^{n-1} \left(\frac{\tau}{\beta * (M-i)} \right) \quad (6)$$

We notice 2 modifications in comparison with equation (4): introduction of β and the addition of a third element in the formula. If we consider a homogeneous network with exponential inter-meeting times statistics, sending a message would be successful only in β proportion of the cases. Thus τ from equation (4) translates into τ/β in equation (6). However, because β intervenes exclusively in the forwarding process, this change affects only the *spray* element in formula (4). The *wait* element, which represents the time needed for nodes to meet their final destination after all copies are sprayed, is unaffected by β since transmission success is assumed at each physical meeting in case of direct delivery.

As compared to non-fragmented forwarding (equation (4)), the fragmented case, *at worst*, needs $n-1$ physical encounters to forward the n^{th} fragment through the network, which is reflected in the third component of the formula (6). That is, the worst case expected latency corresponds to the situation when each fragment is forwarded to a different custodian. This provides the time at which the message can be reconstructed at the destination, supposing erasure coding into n fragments is performed at source.

Equation (6) shows that message latency is affected by both a replication factor L and a redundancy factor n . In order to isolate these 2 effects, and to focus only on the latency component associated with redundancy, we consider the time span δ_n between expected latency with no fragmentation, and expected latency for n fragments. It is easy to show that for a given M and L :

$$\delta_n = T_{sw}(n) - T_{sw}(1) = \sum_{i=1}^{n-1} \left(\frac{\tau}{\beta * (M-i)} \right) \quad (7)$$

δ_n reflects the networking latency overhead due to erasure coding. We will come back to this equation in the evaluation section.

D. Proposed approach

Our goal is to arrive at the right redundancy factor given other network parameters. That is, given replication factor L , the number of nodes M , the number of fragments $m = \lceil s/f \rceil$, mean pairwise meeting time τ , forwarding probability β and time-to-live T , we would like to find an appropriate n such that the message is delivered before T with high probability. Using M, L, τ, β , we can compute the expected latency under worst conditions for a message being erasure coded for different candidate n 's ($1, 2, \dots$). Running a series of computations for $i = 1, 2, \dots$ we get a series of $T_{sw}(i)$ that we can feed into equation (5) with the given T . From the series of $p_T(i)$ thus obtained and the given m , we can find the appropriate n applying Poisson trials theory³ or one of its approximations. Varying this procedure for different L s we obtain a series of (L, k) pairs which produce the same (given) reconstruction probability. Next, we use the cost optimisation function described in the next subsection.

E. Minimising delivery costs

In the previous subsection we have calculated a series of (L, k) pairs such that a sufficient number of different fragments (at least m , to be specific) arrive at the destination before expiration of the message time-to-live T .

What we need to do now is to calculate the system resource costs related to different (L, k) pairs.

A simple assumption would be that the system resources consumed by the protocol are proportional to the aggregate size of the message, considering all copies that can be sprayed before time-to-live T expires. Let $I_f(t)$ be a function⁴ describing the number of copies of fragment f at time t . Since we are using controlled replication, we have one copy at protocol initialisation and L copies after spraying time. Altman et al. have shown [19] that $I_f(t)$ has a sigmoidal shape which is however close to the following linear form:

$$I_f(t) = \begin{cases} L * t/T_s & \text{if } t \leq T_s \\ L & \text{if } t > T_s \end{cases} \quad (8)$$

where T_s is the spray phase time

$$T_s = \sum_{i=1}^{L-1} \left(\frac{\tau}{\beta * (M-i)} \right)$$

As opposed to replication, erasure coding at source brings immediately into the system a number of copies which

³Note that Poisson trials are applicable due to independence of arrival of fragments at destination.

⁴Also known as infected nodes function.

remains constant in time - namely n (immediately, as erasure coding takes place before the first fragment is sent). Let $C(t)$ be a measure of resource consumption for a given message as a function of time. We assume that when T is reached, the system is able to clean those messages with expired time-to-live. This implies that C has its maximum for $t = T$. System optimisation might be attained, in this case, by minimising costs for a known m (resulting from Section III-C), given a number of nodes M , and a time-to-live T before which the entire message should be reconstructed successfully at destination.

Thus, our minimising delivery cost problem can be formulated as follows:

$$\begin{cases} \text{calculate} & (L, k) \\ \text{subject to minimise} & C(T) = I_f(T) * n \end{cases}$$

V. EVALUATION

In this section we aim to validate the model presented in the previous section, by comparing simulation results with those obtained from numerical evaluation as described below. Our objective is (1) to provide evidence that the mathematical model is reasonable compared to simulations, and demonstrate its use, and (2) to present some optimisation results and obtain some insights on the relation between resource cost and protocol performance.

A. Simulation setup

We set up a simulation environment using ONE [22], built upon a delay-tolerant network of $M = 100$ nodes moving according to random waypoint mobility with $\tau = 60$ minutes, with a forward probability modelled by β . Our test scenario consists in injecting 100 messages, each sized 1kB, from every node to a different random node while message time-to-live is set to $T = 9$ minutes. Every message is erasure coded at source and we require at least $m = 10$ fragments delivered in order to reconstruct the message at destination. We have enforced β in the routing layer of ONE, in such a way that only a subset of the message list, corresponding to the given β , was forwarded at each physical meeting. We have chosen random waypoint mobility in order to build up a homogeneous network, given the fact that the mathematical model is also built on this assumption. Every point plotted in the simulation figures is the result of 10 measurements where we varied the initial node position and initial direction of movement.

B. Validation of the model

We recall that equation (7) describes the latency overhead for introducing erasure coding into the network. To validate equation (7) - and consequently equation (6) - we have compared the computations arrived at using equation (7) with different simulation results.

Figure 4 plots the latency overhead with respect to 10 fragments (δ_{10}) from simulations and compares it with the computed latency overhead. It shows that the calculated δ_n is a good upper bound approximation compared to the simulation attempts which are denoted by the crosses on the chart. It appears also that δ_n is in reverse proportion to β .

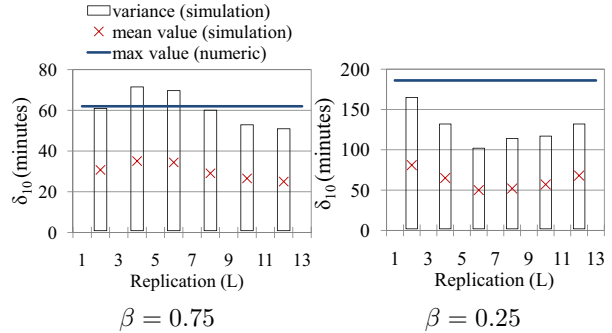


Figure 4. Approximation of mean latency due to fragmentation

Figure 5 shows the result of computing a range of k s given a range of L s using the mathematical model denoted by the numerical curve. We have also constructed simulation trials so that for a given L the latency for the m^{th} fragment ($m = 10$) is within a close margin (1%) to the time-to-live parameter T . Every cross on the figure is the result of 10 such experiments providing a k based on that L . The experiments have been repeated for 2 different β s.

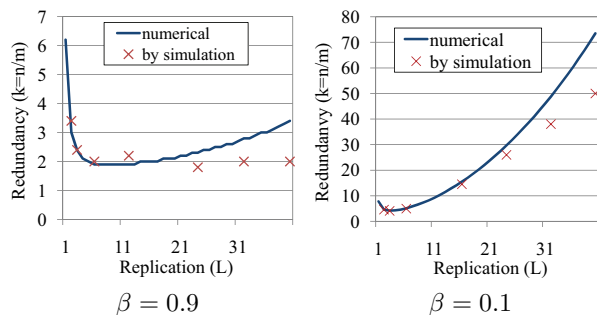


Figure 5. Replication L versus redundancy k

Simulations demonstrate a reasonable match with the numerical results, with closer match for smaller values of L , as well as for smaller values of β . Multiplication of paths may explain why for smaller β s, simulation matches better our numerical results. We recall that, in our model, equation (6) provides expected latency under worst case conditions, when each fragment takes a different path, which is more likely to happen with a smaller β .

Next, we show that network costs, equivalent to the overall number of message copies at time T , represents a meaningful value in the actual economy of the network. We use the same network simulation settings as before and plot in Figure 6 the calculated cost and the actual buffer

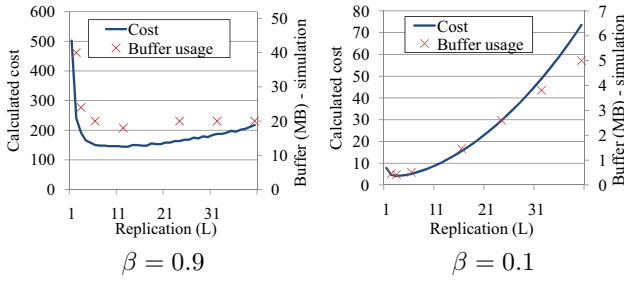


Figure 6. Cost versus buffer occupancy

occupancy after T time units on all nodes, as a function of replication L . These findings also provide a good validation for our theory through simulation.

C. Optimisation results and insights

The practical benefit of the proposed optimisation scheme is that the system calculates the best suited (L, k) pair as a function of the given time-to-live T and the number of fragments required m . Message time-to-live T is the most important parameter because the aggressiveness of the protocol (as well as the cost) is greater when short delivery times are required. In the figures below we show a number of numerical studies varying a number of input parameters.

In Figure 7 we present the cost associated with 4 different time-to-live parameters ranging from a short one ($T = 0.06\tau$) to a long one ($T = 0.5\tau$). We note that, for our network, the minimal cost is achieved for a replication factor generally falling between $L = 4$ and $L = 20$ together with a correlated amount of redundancy.

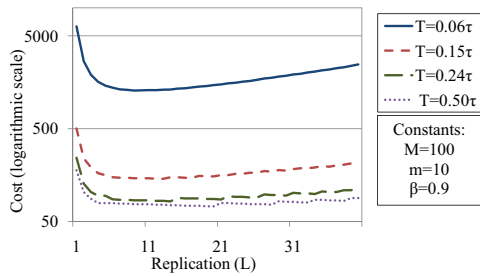


Figure 7. Effects of T on cost

It also appears that redundancy is a practical way to overcome the fact that the replication L is an integer whilst the ideal L may not be one. Especially for long T s, which are associated with low protocol aggressiveness, fractional values of k may be used together with an integer L in order to produce results “similar” to a fractional L .

In Figure 8 we study whether the optimisation of costs would be materially affected by the choice of m which is ultimately influenced by the message size. Supposing that the network allows a maximum fragment size f , we need more fragments for a long message than for a short message

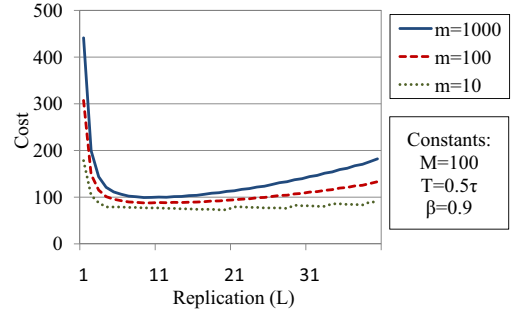


Figure 8. Effects of m on cost

(calculated as $m = \lceil s/f \rceil$). Figure 8 shows that overfragmentation is not beneficial for the cost, as multiplication of paths can delay message reconstruction. Besides, when a high level of fragmentation is used, we have additional drawbacks that are not studied in this paper but include increasing reconstruction effort at destination, additional resources taken up by every fragment header, etc. All these facts speak for a careful correlation between contact window and fragmentation, as proposed in section III-C.

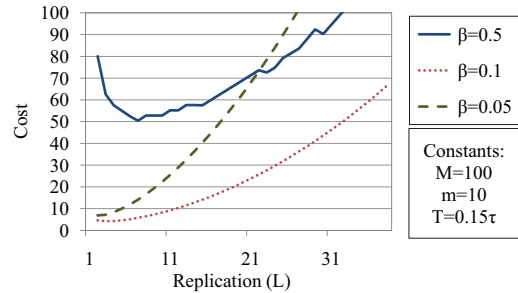


Figure 9. Effects of β on cost

Figure 9 shows how forwarding probability β affects cost evaluation. We note that the best cost is found neither at the bottom end of β nor at the top end, and this might be counterintuitive at first. In the case presented above, best cost is achieved for a $\beta = 0.1$, with a significantly lower cost than for $\beta = 0.05$ and, surprisingly enough, than for $\beta = 0.5$. However, Altman et al. arrived at a similar conclusion [19] by voluntarily diminishing forwarding probability in order to save energy. In our model, it is obvious that, up to a certain point, the number of copies (and implicitly the cost) will diminish when no forwarding opportunity exists. In such overloaded networks, cost minimisation will then favour a high degree of redundancy over a high degree of replication. We note that the L corresponding to the minimum cost decreases from $L = 8$ at $\beta = 0.5$ to a much smaller $L = 2$ corresponding to $\beta = 0.05$. All these examples confirm that the model presented in this paper has enough flexibility to handle a high variety of cases.

VI. ADAPTATION OF THE BASIC MODEL

In this section we propose an improvement of the previous model, considering now a generic distribution of delivery times instead of a slightly imprecise approximation of a logarithmic distribution between bounds, defined by T_o and T_{sw} . This will help us apply the optimisation model to any opportunistic algorithm that has controlled replication and redundancy, as long as some estimates of average latency are available.

The actual distribution of delivery times $l_{i,j,L} = \{T_1, T_2, \dots, T_n\}$ may be harvested from delivery history at each destination j and transmitted periodically to the relevant source i as shown in Algorithm 1 below. Upon arrival at destination, i will then reconstruct the associated probability vector $p_{i,j,L} = \{p_{T_1}, p_{T_2}, \dots, p_{T_n}\}$ where p_{T_n} is the probability that the n^{th} fragment reaches the destination before T expires. Of course, this adds somewhat to overheads but can be seen as a method to extend the reach of the model to new routing algorithms that have the commonality of opportunistic contacts, replication and redundancy. Moreover, it allows a protocol to adapt to the actual network characteristics as long as past history can be assumed to have some bearing on future behaviour. Related overheads can be kept low by allowing information to be sent only when it reaches a threshold size.

In this generic case only the method of calculating probabilities is replaced, but the method of calculating n , as well as the minimisation of cost, still hold. One can also see the generic case as the second (adaptive) phase of a network operation that is initially operated using the basic model.

In the generic case, optimised parameters can be calculated as follows:

- calculate f for the network based on mobility model and expected node velocities (equation (2)),
- for each message, calculate number of fragments to be sent $m = \lceil s/f \rceil$,
- determine ideal n and L corresponding to the minimum cost, according to Subsection IV-E. Assuming that delivery time distribution is transmitted from destination back to source, ideal n and L are only function of message time-to-live T .

VII. CONCLUSION

In this paper we have addressed several aspects of optimisation in the context of a generic, store-carry-forward protocol assuming both controlled replication and erasure coding. We have proposed an equation for contact window estimation that can be useful when we encounter a wide and heterogeneous range of message sizes. This has earlier been used to calculate, at each contact, the maximum transferable message size [6]. We have enhanced this concept by proposing a network-wide constant data unit size, better suited for fragmentation together with erasure coding. The latter

```

ni // source node
nj // destination node
mk // message k sent from ni to nj with
      // replication factor L
li,j,L // list of delivery times from ni to nj
      // using replication factor L
Tk // delivery time of mk from ni to nj
pT // probability that one single message (or
      // fragment) is delivered before time T

// at destination nj
on receiving mk from ni do
  | push Tk → li,j,L;
  | if size(li,j,L) ≥ threshold then
  |   | transmit li,j,L to ni;
  |   | li,j,L → ∅ ;
  | end
end

// at source ni
on receiving li,j,L from nj do
  | merge local li,j,L with remote li,j,L;
end

// computing pT at ni for a known T
pT(L) =  $\frac{\text{countElements}((l_{i,j,L})[k] \leq T)}{\text{size}(l_{i,j,L})}$ 

```

Algorithm 1: pseudo-code: getting p_T from history

is derived from estimations based on the mobility model instead of relying on coordinate and velocity exchange at every contact.

We have proposed a novel mathematical model linking together controlled replication and erasure coding. By analysing the distribution of delivery times, we can calculate the probability for a message to be delivered using replication and erasure coding before its time-to-live expires. By applying minimisation on delivery costs, for each message we can calculate its own optimised replication factor and redundancy factor, taking into account message size and time-to-live.

We have tested the validity of the proposed model by simulations and have demonstrated a good correlation between analytical (calculated) values and those gathered in a simulation environment. We have then shown the effects of the parameters involved: message time-to-live, fragmentation level, and forwarding probability. Our findings have revealed that message time-to-live, forwarding probability, and to a lesser extent message size, dramatically affect the choice of replication/redundancy parameters.

The model was then enhanced from being dependent on the latency bounds calculated for a given protocol (SprayAndWait), to a generic model that adapts itself based on delivery history. In case the destination node keeps a record of delivery times and occasionally sends it to the

source, this information can be used as an input for the optimisation process. Because this mechanism remains valid for an arbitrary protocol that uses replication and erasure coding, when neither upper and lower bounds, nor distribution of delivery times can be analytically determined, it represents an improvement by removing some of the model's simplifying assumptions. Future work includes distribution of the adaptive framework in a simulated environment allowing the study of convergence properties, and adaptation of the analytical models to non-uniform mobility models and other protocols using replication and erasure coding. Another direction of future work would be to make the model aware of energy consumption, and thereby allow the optimisation of parameters in an energy-constrained setting.

REFERENCES

- [1] S. Farrell and V. Cahill, *Delay and Disruption Tolerant Networking*. Artech House, October 2006.
- [2] T. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, CS-2000-06, July 2000.
- [3] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, August 2005.
- [4] —, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 77–90, February 2008.
- [5] —, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 63–76, February 2008.
- [6] G. Sandulescu and S. Nadjm-Tehrani, "Opportunistic Routing with Window-aware Adaptive Replication," in *Proceedings of the 4th Asian Conference on Internet Engineering, ACM*, November 2008.
- [7] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance modeling of epidemic routing," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 51, pp. 2867–2891, July 2007.
- [8] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *Lecture Notes in Computer Science*, vol. 3126, pp. 239–254, January 2004.
- [9] Z. J. Haas and T. Small, "A New Networking Model for Biological Applications of Ad Hoc Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 27–40, February 2006.
- [10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," in *Proceedings of the 5th IEEE International Conference on Pervasive Computing and Communications Workshops*, March 2007.
- [11] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss., "Delay-tolerant networking architecture (RFC4838)," April 2007. [Online]. Available: <http://www.ietf.org/rfc/rfc4838.txt>
- [12] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network," *ACM SIGCOMM Computer Communication Review*, vol. 35, pp. 109–120, October 2005.
- [13] J. Widmer and J.-Y. L. Boudec, "Network Coding for Efficient Communication in Extreme Networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, August 2005.
- [14] M. Pitkänen and J. Ott, "Redundancy and Distributed Caching in Mobile DTNs," in *Proceedings of the 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, August 2007.
- [15] G. Sandulescu and S. Nadjm-Tehrani, "Adding Redundancy to Replication in Window-aware Delay-tolerant Routing," *Journal of Communications, Special Issue in Delay Tolerant Networks, Architecture, and Applications, Academy Publisher*, vol. 5, no. 2, pp. 117–129, March 2010.
- [16] J. Ott and D. Kutscher, "A disconnection-tolerant transport for drive-thru Internet environments," in *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM05)*, vol. 3, March 2005.
- [17] M. Pitkänen, A. Kernen, and J. Ott, "Message Fragmentation in Opportunistic DTNs," in *Proceedings of International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2008)*, June 2008.
- [18] R. Groenvelt, P. Nain, and G. Koole, "The message delay in mobile ad hoc networks," *Elsevier Journal of Performance Evaluation*, vol. 62, pp. 210–228, October 2005.
- [19] E. Altman, T. Başar, and F. D. Pellegrini, "Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks," *Elsevier Journal of Performance Evaluation*, vol. 66, pp. 1–19, September 2009.
- [20] Y. Lin, B. Li, and B. Liang, "Efficient Network Coded Data Transmissions in Disruption Tolerant Networks," in *Proceedings of 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM08)*, April 2008.
- [21] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM04)*, Venice, Italy, October 2004.
- [22] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools09)*. New York, USA: ICST, March 2009.