

# Vicinity Resource Cartography for Delay-Tolerant Networks: a Holistic Perspective

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

Peter Schaffer  
University of Luxembourg  
peter.schaffer@uni.lu

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

**Abstract**—In this paper we propose a distributed approach to construct vicinity resource maps at each node in a delay-tolerant network. The scheme uses recent encounters to estimate the availability of the following resources: energy, buffer space, and bandwidth, in the vicinity of a given node. We then show how a store-carry-forward scheme may benefit from having access to these resource estimates. While knowledge about resources available in the vicinity allows nodes to implement meaningful custodian election strategies, an evaluation of available bandwidth and contact time allows them to carry out adaptive queue management strategies. Thus, such strategies can be approached from a holistic perspective based on the availability of the three resources under consideration in node proximity. We validate our model separately in time-varying and space-varying environments. In addition to synthetic mobility models (random waypoint) we validate our resource estimation in a disaster area mobility model (using Bonn motion traces). We show that by using this information a routing protocol may dramatically improve its delivery rate and reduce energy overhead while keeping delivery latency almost constant.

**Index Terms**—routing; resource management; delay-tolerant networks

## I. INTRODUCTION

Opportunistic routing schemes are designed to deliver messages in the absence of any scheduling information in partitioned networks. The latter have often been referenced in the literature as delay- and disruption-tolerant networks (DTN) [1]. The idea is to allow some network nodes to participate in forwarding messages in partitioned ad-hoc networks using the store-carry-forward paradigm. Many opportunistic schemes assume that nodes possess an average level of network resources that are homogeneously distributed over the network [2], [3].

However, resource distribution is not homogeneous in real networks, so the average resource assumption usually does not hold. Real networks may be composed of handhelds, car embedded computers, laptops, fixed throwboxes and sensors, which obviously have different energy requirements and different buffer space allowances. Moreover, neither message source nodes, nor message destination nodes need to be uniformly distributed across the node population. Therefore network resource consumption is uneven. Given this resource heterogeneity, having a local estimate for the resources available in the vicinity of each node would be highly beneficial for routing decisions. For example, based on such estimates a node may decide to whom to forward its messages (custodian election)

and which messages should be transferred first (queue management).

In an attempt to enhance delivery performance and to minimise end-to-end delay in disconnected networks, recent studies have looked at node behaviour from several new perspectives. Some put forward the idea of social correlations between nodes which may influence node contact distribution [4], while others propose techniques based on a combination of social context and node topology information [5], [6]. These techniques select some key nodes, among those that are located more centrally in the network or are more active socially, considering them to be better suited for forwarding incoming messages. In so doing, however, they place additional strain on the resources available to these key nodes, and they will be the first to fail as a consequence of network congestion, buffer exhaustion, or battery depletion. Thus, in the absence of any metrics for the resources available to nodes, these strategies may actually jeopardise initial performance gains by losing the most important nodes through overuse.

Other studies focus on the particular movements building up higher densities around points of interest [7]. However, in partitioned ad-hoc networks the localisation of such points of interest, acting as resource concentrators, is difficult to achieve. Moreover, resource concentrators may move in space and vary in time. For example, the distribution of resources in a disaster area cannot be planned in advance. Somewhat similarly, the configuration of points of interest in case of a traffic jam triggered by an accident varies in space and time as an effect of the accordion phenomenon [8]. Finally, even when the geographical distribution of resources is known in advance and is constant in time, being able to detect them autonomously, that is, without central knowledge, makes a big difference in terms of protocol robustness.

Knowing the time-varying and space-varying resource distribution in a network can have a huge impact on the choice of routing strategies, even when only *approximate* knowledge is available. An important problem then is how to select custodians efficiently, i.e. depending on own resources and on resources available in the neighbourhood. For example, a node may choose not to forward a message at a particular encounter knowing that better opportunities may show up in the near future. Another problem to consider is how to prioritise messages in a message queue so as to give them a good chance to be transferred within the limited contact

window of an encounter.

In this paper we propose a distributed algorithm for estimating the resources available in the proximity of a node, with no a priori knowledge. We study the accuracy of this distributed algorithm and validate it in three different simulation settings. We consider separately space-varying and time-varying resource maps, and perform validation in both a random waypoint scenario and a disaster area scenario (using Bonn motion traces [9]). We then go on to propose a set of policies where those estimates are used for selecting a suitable custodian (custodian election) and for choosing the right message to transfer (queue management). By applying these policies in the context of a standard Epidemic protocol [10] we show substantial performance enhancement as compared to the standard routing baseline.

To sum up, the contributions of this paper are as follows: (1) We propose a distributed algorithm for calculating resource availability in node vicinity; (2) We validate the algorithm in different scenarios, showing that its inaccuracy is below 10% in the scenarios considered; (3) We propose and then implement custodian election and message queue strategies using all three resources in a holistic way. We demonstrate a 30% gain in delivery rate and a 22% gain in energy consumption as compared to a standard Epidemic routing baseline in a disaster area scenario.

## II. RELATED WORK

To the best of our knowledge, there is no unified proposal in the literature linking together the most relevant resources in DTN: energy, buffer space, and bandwidth, into one single abstraction. In our case, past encounters are used to predict the evolution of available resources in the near-term. Prediction-based schemes have already been proposed in earlier works [3], [11], but they mostly deal with contact probability, thus ignoring the amount of resources nodes are contributing to the network. On the other hand, in a heterogeneous environment, where resources range over a wide spectrum of types and levels, estimating resource availability remains an open question.

Another prediction-based protocol, MobySpace [12], proposes to construct a high-dimensional Euclidean space from node mobility patterns, while also assuming prior knowledge (or learning) of those patterns. We have also used Euclidean space in our approach, but we have focused on the resources available in node vicinity. Moreover, the equations we have proposed are not tightly constrained by one mobility model or another, as demonstrated by the use of two alternative scenarios.

Heterogeneity has been studied by Spyropoulos et al. [13], who propose a utility function for the selection of appropriate custodians. However, their paper does not consider accounting for resources in general, or bandwidth in particular, which leads to the questionable assumption that a message may be conveyed over a meeting regardless of its size.

## III. RESOURCE AVAILABILITY IN PARTITIONED NETWORKS

In this section we propose an algorithm for calculating the level of resources available in the vicinity of a node. We start by developing a generic model, which we then apply to individual network assets, such as buffer space, energy, and bandwidth.

We consider a sparse network where resources are potentially not uniformly distributed. As a consequence, there may be pockets of resources in the network, in the form of energy or buffer space, which may vary both in time and space. Incidentally, the way in which resources are distributed in a network may also be the consequence of mobility, but we are not assuming any particular mobility pattern for our model. As an effect of sparsity, we will assume that most meetings happen between two nodes.

### A. Resource variation in time and space. The generic model

We consider a generic resource  $R$  for a given node. Obviously, every node knows its own resource level. Let us denote with  $R_A^o$  the own resource level available at node  $A$  and with  $R_A^v$  the estimated resource available in the vicinity of node  $A$ . Applying the algorithm below, every node will be able to evaluate the expected virtual value of resource  $R$  available in its vicinity.

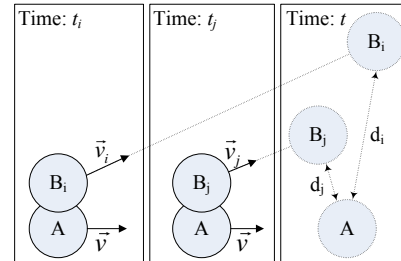


Fig. 1. Node  $A$  meetings history within observation time span

Let us consider a node  $A$  moving at speed  $\vec{v}$ , and a set of nodes  $B_i \dots B_j$  moving at speeds  $\vec{v}_i \dots \vec{v}_j$ , respectively. Every node knows its own vectorial speed and resource level, and communicates them at each meeting. For instance, when a meeting happens between node  $A$  and node  $B_i$ , node  $A$  receives vector  $I_i$  from node  $B_i$ :

$$I_i = (\vec{v}_i, R_i^o)$$

Consider a sequence of encounters between node  $A$  and nodes  $B_i, \dots, B_j$  at times  $t_i, \dots, t_j$  at which  $A$  collects resources  $R_i^o, \dots, R_j^o$  from those nodes. We assume that  $A$  maintains a log of maximum  $j - i + 1$  encounters during a sliding time window of size  $\tau$ . After meeting the last node ( $B_j$ ), node  $A$  will have the following information available:

$$M_A = \begin{pmatrix} t_i & t_{i+1} & \dots & t_j \\ \vec{v}_i & \vec{v}_{i+1} & \dots & \vec{v}_j \\ R_i^o & R_{i+1}^o & \dots & R_j^o \end{pmatrix}$$

$M_A$  is determined by adding local time information  $t_k$  to the vector  $I_k$  received at each meeting with node  $B_k$ . We note that the information the node needs to store is limited for two reasons: first, because the maximum number of columns of the matrix ( $j - i + 1$ ) is limited, and second, because all information older than  $\tau$  is discarded (such as  $t_j - t_i < \tau$  at any point in time).

Considering that speeds  $\vec{v}$  and  $\vec{v}_i, \dots, \vec{v}_j$  remain constant for the short observation time span  $\tau$ , we can calculate the Euclidean distance between node  $A$  and nodes  $B_i, \dots, B_j$  at a later point in time than the encounter time, as shown in Figure 1. We can also calculate the contribution of node  $B_k$  to the estimated virtual resources of  $A$  as being in reverse proportion to the distance from  $B_k$  to  $A$ . We denote the resource footprint in the vicinity of  $A$  as  $R_A^v$ , and compute it as a function of resources met by node  $A$  during the time span  $\tau$ . Finally,  $R_A^v$  is proportional to the number of meetings  $A$  had over the observation time  $\tau$ . As we do not want to limit our findings to one particular mobility model, we do not use meeting probabilities. Consequently, we need to rely on some derived measures. Putting all the information together, we can apply the formula below to calculate the estimated available resource in the vicinity of node  $A$ :

$$R_A^v = \underbrace{\frac{n_\tau}{\omega \times \tau}}_{c_A} \times \frac{\sum_{k=i}^j \frac{R_k^o}{d_k}}{\underbrace{\sum_{k=i}^j \frac{1}{d_k}}_{\bar{R}_A}} \quad (1)$$

where:  $\tau$  = observation time span  
 $\omega$  = node's average meeting frequency  
 $n_\tau$  =  $j - i + 1$ , number of nodes  $A$  actually met during observation time  $\tau$

In other words, as meetings take place and resource information is exchanged, each node builds up its own map of virtual resources, assigning greater weight to those at a shorter distance.

Equation (1) can be decomposed into:

- an element  $c_A$  reflecting the density of meetings in the given region. This acts as a generic factor irrespective of resource type  $R$ .
- an element  $\bar{R}_A$  representing the average availability of resource  $R$  weighted by the inverse of the distance between  $A$  and the nodes met.

The validation of this formula is done in extensive scenarios in Sections IV and V but here we consider the following special cases for discussion:

- if node  $A$  had no meetings over the  $\tau$  time span:  $\lim_{n_\tau \rightarrow 0} R_A^v = 0$  because  $n_\tau$  respectively  $c_A$  is 0
- if the node actually had an average number of meetings over time span  $\tau$ :  $\lim_{c_A \rightarrow 1} R_A^v = \bar{R}_A$

## B. Modelling individual network assets

After having developed a generic model for calculating resources available in node vicinity, let us now move on to the second step in our modelling exercise, and refine this equation for individual network assets: buffer space, energy, and bandwidth. While buffer space and energy are properties related to one node, bandwidth is a property linking together two or more nodes. We treat the three categories of resources in an increasing order of complexity.

1) *Buffer space*: The buffer space case is straightforward. Equation (1) can be used directly for calculating buffer space by simply replacing the generic resource with buffer space in the formula. This is possible because buffer space remains constant as long as no messages are exchanged between nodes. That is, for short observation times  $\tau$  and low network load, we can approximate a node's buffer space at time  $t$  with the buffer space we have observed at time  $t_i > t - \tau$ . Denoting the available buffer size with  $S$ , we can directly replace  $R$  by  $S$  in Equation (1).

2) *Energy*: The energy model is more complex, because energy levels do not remain constant, even in the absence of message exchange. In case there is traffic, energy is depleted by the sending and receiving of messages at a rate approximately proportional to the size of messages exchanged. In case there is no traffic, node energy decreases simply due to network sensing. We can approximate the energy level at a node, at one particular timepoint  $t$ , by relating it to the relevant factors, as follows:

$$E^o(t) = E_{max} - \underbrace{e_s \times t}_{\text{Energy for sensing}} - \underbrace{e_m \times m}_{\text{Energy for message exchange}}$$

where:  $E^o(t)$  = node's own energy at time  $t$   
 $E_{max}$  = maximum energy available for this type of node  
 $e_s$  = energy factor for sensing  
 $e_m$  = energy factor used for message exchange  
 $m$  = total size of exchanged messages

Factors  $e_s$  and  $e_m$  can both be measured for different types of nodes in a laboratory setup [14]. In a simpler setup, every node can measure energy depletion as a function of time just by retrieving battery levels at 2 different times. Denoting this attenuation rate by  $e$ , the above equation is simplified as follows:

$$E^o(t) = E_{max} - \underbrace{e \times t}_{\text{attenuation with time}}$$

Thus, time variable  $E^o(t)$  replaces the constant  $R_k^o$  in Equation (1) as a node's estimate for own energy.

3) *Bandwidth*: For the purposes of this model, we define bandwidth as the maximum volume of data  $D_N$  that a node  $N$  can exchange at one meeting. Our model is meant to achieve a twofold objective: first, to provide an estimate for a node's capacity to send and receive messages (at one meeting or over a given time span); and second, to help determine which message to send, depending on message size and the estimated probability of success.

The theorem we propose below will allow us to estimate the volume of data exchanged between two nodes, as well as the probability of a message to pass, taking contact window estimation as a basic factor. Contact window denotes the time during which two nodes are in radio range of each other, and represents a critical factor for realistically evaluating bandwidth in mobile networks. As mentioned earlier, we consider that most encounters will happen between only two nodes, as a result of network sparsity.

*Theorem 1:* For a meeting between two nodes (disk radio range with radius  $r$ ), moving at a relative speed of  $\vec{v}_{rel}$  and communicating over a protocol with nominal bandwidth  $b_n$  we can calculate:

- (I) the maximum volume of data exchanged during the meeting as:

$$D_{max} = \frac{2r}{|\vec{v}_{rel}|} \times b_n$$

- (II) the expected volume of data exchanged during the meeting as:

$$D_{exp} = \frac{\pi r}{2|\vec{v}_{rel}|} \times b_n$$

- (III) the probability of an exchange exceeding a given size:  $Pr\{D \geq pD_{max}\} = \sqrt{1-p^2}$  where  $p \in [0, 1]$

*Proof:* As shown in Figure 2, a node crosses the radio range of another on a trajectory  $\overline{PQ}$ . This trajectory is covered at a speed  $\vec{v}_{rel}$  and  $P$  is the incidence point between the nodes' radio ranges. Using geometry we can calculate the trajectory between  $P$  and  $Q$  as a function of  $x$ :

$$f(x) = \overline{PQ} = 2\sqrt{x(2r-x)}$$

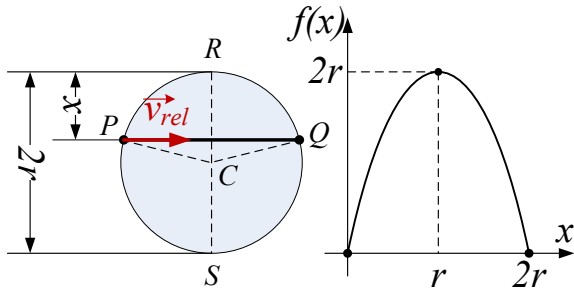


Fig. 2. Contact window as a function of incidence point

If we start out from the assumption that nodes meet,  $x \in [0, 2r]$  and contact point  $P$  may be anywhere on the circle arc  $\overline{RPS}$ . Moreover, in the generic case,  $x$  is a random variable uniformly distributed over the interval  $[0, 2r]$ . Thus,  $D_{max}$  can be calculated as:

$$D_{max} = \frac{\overline{PQ}_{max}}{|\vec{v}_{rel}|} \times b_n = \frac{2r}{|\vec{v}_{rel}|} \times b_n \quad (\text{I})$$

$$D_{exp} = \frac{\overline{PQ}_{exp}}{|\vec{v}_{rel}|} \times b_n = \frac{\int_0^{2r} f(x) dx}{|\vec{v}_{rel}|} \times b_n = \frac{2r^2 \arctan\left(\frac{\sqrt{x}}{\sqrt{2r-x}}\right) + \sqrt{x(2r-x)}(x-r)\big|_0^{2r}}{2r|\vec{v}_{rel}|} \times b_n = \frac{\pi r}{2|\vec{v}_{rel}|} \times b_n \quad (\text{II})$$

Now, we calculate the probability of having an exchange exceeding a given fraction  $p \in [0, 1]$  of the maximum  $D_{max}$  as:

$$Pr\{D \geq pD_{max}\} = Pr\{f(x) \geq p2r\} = Pr\{2\sqrt{x(2r-x)} \geq 2pr\} = \frac{|x_1 - x_2|}{2r} = \sqrt{1-p^2} \quad (\text{III})$$

where  $x_1$  and  $x_2$  are the solutions of the quadratic equation  $2\sqrt{x(2r-x)} = 2pr$  ■

Theorem 1 allows us to estimate one particular network asset – bandwidth – characterised at one node, namely the expected data volume that this node can exchange at one particular meeting. We can now easily integrate bandwidth into our model for generic resources in order to estimate the expected bandwidth value in the vicinity of a node, as follows. At each meeting time  $t_i, \dots, t_j$  a node knows its own speed as well as that of its neighbours, so it can calculate the relative speeds  $v_{rel,i}, \dots, v_{rel,j}$ . In a given scenario,  $r$  and  $b_n$  are approximated as known constants. Therefore, by using Theorem 1.II, each node can calculate the array of expected data to be exchanged ( $D_{exp,i}, \dots, D_{exp,j}$ ) and, by using Equation (1), it can estimate the expected amount of data  $D_A^v$  that can be exchanged by node  $A$  over the observation time  $\tau$ .

One additional application of Theorem 1 is that, if node  $A$  actively manages its message queue, it can also choose which message to send by evaluating the probability of successful message transmission given by Theorem 1.III.

#### IV. ALGORITHM VALIDATION

In this section we demonstrate how the virtual resources in the neighbourhood of a node can be estimated by applying Equation (1) upon exchanges between nodes, as proposed in Section III. The goal of validation is to show that these estimates are indeed close enough to the real resource levels in the network. We have organised our validation exercise in two sections: in this section we validate our algorithm applied to buffer space in three different scenarios, while in Section V we validate our holistic approach considering all three network resources.

##### A. Validation in a space-varying environment

As shown by Hyytiä et al. [15], when nodes move according to random waypoint mobility in a square, node density is

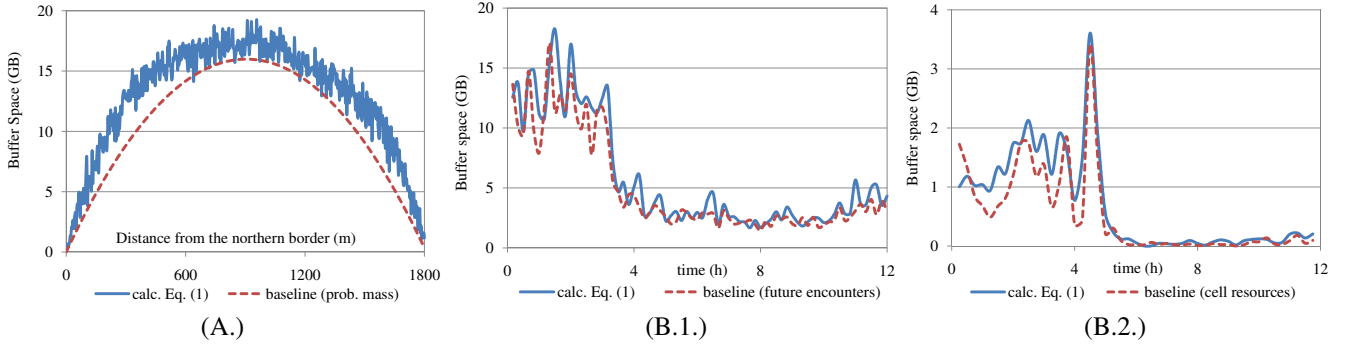


Fig. 3. Calculation versus baseline in a simulation environment; we plot available buffer space in the vicinity of a node

maximum in the middle of the area and decreases to 0 towards the borders. This observation allows us to validate Equation (1) in a simple, yet revealing experiment where we can isolate the time-varying element from the space-varying element. Assuming constant buffer space per node (i.e. nodes do not exchange large messages, but only small amounts of information, as required for Equation (1)), virtual buffer distribution is given predominantly by node densities and is constant in time.

We used the ONE simulation environment [16] where we considered a set of 100 nodes moving at 20 m/s according to random waypoint mobility within a space defined by a square with an edge length of 1800 m, as well as a "spy" node  $A$  moving very slowly at 0.04 m/s on a rectilinear path starting from the middle of the northern edge and ending in the middle of the southern edge. Initially, all nodes have a fixed amount of buffer space (500 MB) and, since no message is delivered between nodes, the resource map is defined exclusively by node density. We modelled the random waypoint movement by creating 100 different movements and computing the average value of  $R_A^v$  over time, according to Equation (1). We considered an observation period of  $\tau = 5$  minutes over a 12-hour scenario, while limiting the  $M_A$  columns to 40. In Figure 3 (A.) we plot the values calculated for  $R_A^v$  (as recorded by the "spy" node  $A$  using Equation (1) for buffer space) versus the values given by the baseline (node density probability mass computed according to the formula presented in [15]), for the same "spy" node. As can be seen in Figure 3 (A.), Equation (1) closely follows the baseline values for buffer space in the vicinity of the "spy" node. An explanation for the slight difference between the baseline and the calculated values may be that while baseline values consider a 0 radio range (calculating only node densities), values calculated using Equation (1) consider a radio range of 20 m.

### B. Validation in a space- and time-varying environment

In the previous subsection, the spatial distribution of resources provided us with a simple baseline. However, if we move on to a space- and time-varying model, or a non-synthetic mobility model, choosing a baseline becomes more complicated. In this subsection we propose two scenarios, with two different baseline alternatives:

- **future encounters**, calculated as the sum of own resources ( $R^o$ ) of all peers that will be actually met by the observed node over a reference timespan<sup>1</sup>  $\tau_f$ .
- **cell resources**, calculated as the sum of own resources ( $R^o$ ) of all nodes sharing the same cell as the node itself at a given time. (Cells are obtained by dividing the simulation playground into a number of equal squares.)

1) *Random waypoint scenario*: In this scenario, our setting was again a 100-node network performing random waypoint movement in the 1800 m  $\times$  1800 m square playground. We then injected a large number of messages over a short period of time in this network, which reduced buffer space in most nodes to a minimum level. This was followed by a period of slow recovery in buffer space, as the messages were gradually delivered and therefore deleted from the buffers.

Figure 3 (B.1.) shows the evolution in time of  $R_A^v$  for a representative node, using buffer space data, as compared to the future encounters baseline. However, comparable accuracy can be found for all the other 99 nodes. If we define inaccuracy  $I$  as the distance between the calculated curve  $c(t)$ , and the baseline curve  $b(t)$ , we can come up with the following formula:

$$I = \frac{\int_0^T |c(t) - b(t)| dt}{\int_0^T b(t) dt} \quad (2)$$

where  $T$  is the simulation time (12h). By applying this formula, we find that inaccuracy  $I$  is between 2% and 6% for all 100 nodes in this scenario, for both baselines proposed: future encounters and cell resources.

2) *Disaster area scenario*: We consider a 150-node network moving according to a disaster management scenario as described earlier [9], known as Bonn Motion. In order to create some heterogeneity in the system, nodes were divided into 3 groups, each including 50 nodes. Only the first group of nodes were injecting messages over the first half of the 12 h scenario, addressed uniformly to all nodes. Buffer space allocation was also uneven: it was 500 MB for the first two groups of nodes

<sup>1</sup>For the sake of simplicity we will use  $\tau_f = \tau$

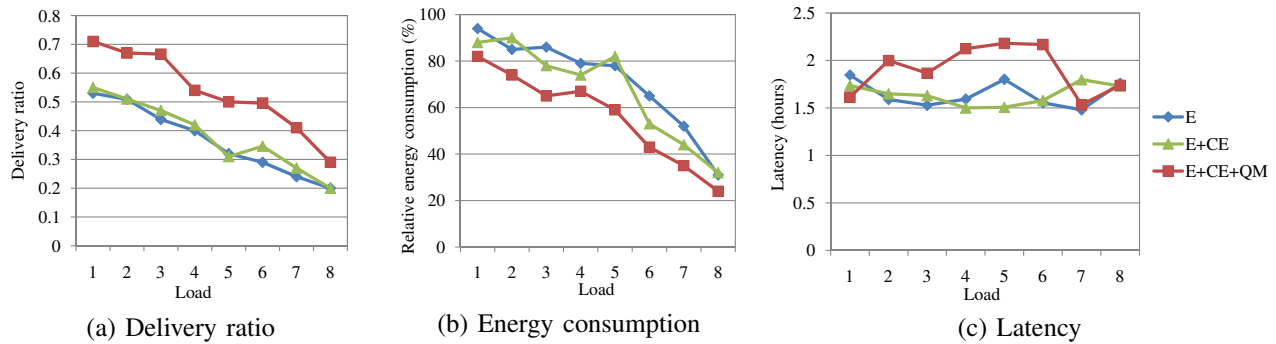


Fig. 4. Performance comparison of plain Epidemic routing (E) with Custodian Election (E+CE) and Queue Management (E+CE+QM)

and only 50 MB for the third one. Playground was  $360 \text{ m} \times 170 \text{ m}$  and the cell used to calculate the cell resource baseline was  $10 \text{ m} \times 10 \text{ m}$ .

In Figure 3 (B.2.) we present the evolution of  $R_A^v$  for one particular node, calculated for buffer space, compared with the cell resources baseline. For this scenario, maximum inaccuracy as compared to cell resource baseline is 10% and typically below 6%.

We have now demonstrated that the generic notion of virtual vicinity resource can be used in two mobility contexts to estimate the buffer resource with a low level of inaccuracy. We have done this considering both time-varying parameters (load and movement) and space-varying parameters (movement).

## V. APPLYING THE HOLISTIC APPROACH

In Subsection III we proposed the idea that every node can estimate the resource level available in its vicinity by keeping a small matrix  $M_A$  derived from the history of its previous encounters. These were shown to be quite accurate for estimating available resources in the vicinity. Whenever a reference node  $N$  meets a node  $A$ , information about the resources in  $A$ 's vicinity, such as energy ( $E_A^v$ ), buffer space ( $S_A^v$ ) and bandwidth ( $D_A^v$ ) becomes available to  $N$ .

Our goal is to show how this information can be potentially used to optimise the store-carry-forward scheme in intermittently connected networks. Since our strategy is based on an analysis of information available about three resources, we have called our approach holistic. We propose here custodian election and message queue management policies, as elements of strategy that can contribute to improving overall network performance.

### A. Custodian election (CE) and message queue management (QM) policies

In a routing or dissemination context, in order to increase the probability of successful message delivery, nodes need to choose the right custodians in the network. For example, choosing a custodian with insufficient amounts of energy may lead to imminent battery depletion, causing not only loss of connectivity in the network, but also loss of all the messages currently stored in that node's buffer. This situation can be improved by applying our custodian election policy, provided

that custodian candidates signal to elector nodes the amount of resources found in their vicinity, as well as their own resource level. By comparing these two values – own resource versus resource level available in the vicinity – a node will choose node  $A$  as a custodian if and only if  $A$ 's own energy, available buffer space and bandwidth are relatively abundant as compared to the resources expected in  $A$ 's vicinity. The idea is for the nodes to choose a custodian when the chances of them finding a better opportunity are small.

On the other hand, signalling the amount of resources expected to be available in the neighbourhood of a node may also improve message queue management. It is highly important for a node to choose the right-sized message from its message queue in order to actually be able to convey it over the estimated contact window. Starting to transmit a message whose size is uncorrelated with the contact window may increase the risk of partial transmission. Partial transmissions are unwanted phenomena as they imply retransmissions, thus wasting bandwidth and increasing energy requirements [17]. Let us then formalise these policies as follows:

CE) When encountering a node  $A$  in the neighbourhood, elect  $A$  as custodian only if  $E_A^o/E_A^v$ ,  $S_A^o/S_A^v$  and  $D_A^o/D_A^v$  are above a threshold value

QM) Send message  $m$  of size  $s$  only if probability to be conveyed over the contact window - as calculated by Theorem 1.III - is above a threshold value

### B. Performance measurements

In order to establish the end-to-end impact of the policies presented in Section V-A, we have implemented them on top of the Epidemic routing protocol [10]. In our simulation environment, we compared the plain Epidemic protocol (E) first with the same protocol together with the Custodian Election policies (E+CE), and then with the Queue Management policies (E+CE+QM) added to the scheme. We used Bonn motion mobility traces and the same network configurations as described in Section IV-B2 above. In order to validate our approach at different loads, we gradually increased the number of messages sent by a factor of 1 to 8, and plotted the results in Figure 4. This figure shows that the implementation of

these policies can yield significant benefits: while delivery rate increases by 30%, and energy consumption decreases by 22%, there is only a 10% increase in latency.

### C. Costs

A legitimate question concerns the costs incurred (i.e. added overhead) for information exchange between nodes. Costs can be approximated as follows:

- **network costs.** In order to implement custodian election and queue management policies, custodian candidates need to send  $E_A^o/E_A^v$ ,  $S_A^o/S_A^v$  and  $D_A^o/D_A^v$  at each meeting. For calculating the resource level available in the vicinity ( $E_A^v$ ,  $S_A^v$  and  $D_A^v$ ), nodes need to exchange the vector  $I = (\vec{v}, E^o, S^o)$  at each meeting. Considering that each scalar value is expressed by 2 bytes and each vectorial value by 4 bytes we can consider that data exchanged by one node at each encounter is 14 bytes.
- **storage costs.** Each node stores a matrix sized  $n_r \times size_I$ . This translates into storage requirements of 320 bytes in the current setting.

These are simple metrics that can be easily compared with aggregated network workload (for all nodes and the whole duration of the experiment) and buffer size. In our settings, aggregated network workload was about  $10^6$  times above network costs and node average buffer was about  $10^6$  times above storage costs.

## VI. CONCLUSION AND FUTURE WORK

In this paper we have proposed a distributed scheme allowing nodes in a partitioned (DTN) network to estimate the energy, buffer space, and bandwidth levels available in their vicinity. We have tested the validity of the proposed model by simulations and have demonstrated a good accuracy of values calculated using the proposed scheme as compared to different baselines.

We have proposed an optimisation approach that uses this information for custodian election and message queue management policies. These policies may be easily adapted to different store-carry-forward protocols. We have then demonstrated that the benefits of these policies are substantial when combined with an Epidemic routing baseline in a disaster management scenario. We believe that our preliminary findings and proposed solutions may show benefits particularly in networks with heterogeneously distributed resources.

Future work includes refinements of the policies by integrating delivery constraints related to message time-to-live versus estimated delivery latency. It would be interesting to analyse how tuning the different parameters used in the algorithm for resource estimation as well the threshold values for custodian election and queue management may further improve the results shown in Section V. Another direction of future work would be validating this collaboration scheme in a real-life delay-tolerant network.

## REFERENCES

- [1] K. Fall and S. Farrell, "DTN: an architectural retrospective," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 828–836, June 2008.
- [2] 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.
- [3] 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.
- [4] E. Yoneki, D. Greenfield, and J. Crowcroft, "Dynamics of Inter-Meeting Time in Human Contact Networks," in *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Data Mining*, July 2009.
- [5] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8<sup>th</sup> ACM International Symposium on Mobile Ad Hoc Networking and Computing*, September 2007.
- [6] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing," in *Proceedings of the 29<sup>th</sup> IEEE International Conference on Computer Communications*, March 2010.
- [7] M. Piórkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "A Parsimonious Model of Mobile Partitioned Networks with Clustering," in *Proceedings of the International Conference on Communication Systems and Networks*, January 2009.
- [8] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. de Amorim, and J. Whitbeck, "The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing," in *Proceedings of the 28<sup>th</sup> IEEE INFOCOM*, April 2009.
- [9] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini, "Modelling Mobility in Disaster Area Scenarios," in *Proceedings of the 10<sup>th</sup> International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, October 2007.
- [10] T. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, CS-2000-06, July 2000.
- [11] Q. Yuan, I. Cardei, and J. Wu, "Predict and Relay: An Efficient Routing in Disruption-Tolerant Networks," in *MobiHoc '09: Proceedings of the 14<sup>th</sup> ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2009.
- [12] V. Conan, J. Leguay, and T. Friedman, "Evaluating Mobility Pattern Space Routing for DTNs," in *Proceedings of the 25<sup>th</sup> IEEE INFOCOM*, April 2006.
- [13] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in Delay Tolerant Networks Comprising Heterogeneous Populations of Nodes," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, March 2009.
- [14] A. Rice and S. Hay, "Decomposing power measurements for mobile devices," in *Proceedings of the 8<sup>th</sup> IEEE Conference on Pervasive Computing and Communications*, April 2010.
- [15] E. Hyytiä, P. Lassila, and J. Virtamo, "Spatial node distribution of the random waypoint mobility model with applications," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 680–694, June 2006.
- [16] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *SIMUTools '09: Proceedings of the 2<sup>nd</sup> International Conference on Simulation Tools and Techniques*, 2009.
- [17] 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*, vol. 5, no. 2, March 2010.