Reliable Transport in Delay-Tolerant Networks With Opportunistic Routing

Lucile Sassatelli, Arshad Ali, Manoj Panda, Tijani Chahed, Eitan Altman

Reliable Transport in Delay Tolerant Networks with Opportunistic Routing

Lucile Sassatelli, Arshad Ali, Manoj Panda, Tijani Chahed and Eitan Altman

***Abstract*—This article tackles the issue of reliable transport in Delay-Tolerant mobile ad hoc Networks (DTN), that are operated by some opportunistic routing algorithm. We propose a reliable transport mechanism that relies on ACKnowledgements (ACK) and coding at the source. The various versions of the problem depending on buffer management policies are formulated, and a fluid model based on mean-field approximation is derived for the designed reliable transport mechanism. This model allows to express both the mean file completion time and the energy consumption up to the delivery of the last ACK at the source. The accuracy of this model is assessed through numerical simulations, and a detailed investigation of the impact of the system parameters on the performance is conducted. We eventually present a joint optimization of the mean completion delay with or without an energy constraint, so as to identify the optimal set of parameters to use.**

***Index Terms*—reliable transport, mobile ad hoc networks, delay tolerant networks, random coding, fluid models**

I. Introduction

Mobile Ad hoc Networks (MANET) aim at allowing communication between mobile users without any infrastructure. The last NSF report on future directions on wireless networking [2] points out (in page 5) that new wireless networked devices, such as mobile phones, must be able to " facilitate response and rescue operations in disaster areas. [...] these devices must be able to operate even when the infrastructure is disabled or under heavy network demand." These networks are hence meant at complementing infrastructure wireless networks, either to offload or to temporarily replace them. If the spatial density of a MANET is low, then end-to-end communication between a source and a destination is limited by the lack of connectivity. Thus, in order to exchange packets, two mobile nodes must come into the radio range of each other. Owing to the intermittent connectivity, the nodes must rely on the Store-Carry-and-Forward paradigm which inherently entails a delay for communication. That is the reason why such sparse MANET are referred to as Delay Tolerant Networks (DTN). Thus, in order to decrease the transmission delay, a source of traffic has to rely on the mobility of other nodes which act as relays, and takes advantage of the transmission opportunities which occur when the mobile relays come into contact. This forwarding strategy is known as opportunistic routing.

When the source needs to ensure that all the packets it sent made it successfully to the destination, we refer to reliable data transfer. TCP is by far the most deployed protocol for reliable data transfer. However, some effects of wireless channels, such as throughput degradation, capture and unfairness, are even more exacerbated in MANETs [3], [4]. TCP performance degradation in MANETs is due to the fact that TCP is unable to distinguish between losses due to route failures and losses due to network congestion. In order to face this issue, a number of proposals have been proposed in the literature. We point out the differences between these proposals and our present in the next Related works section.

Random network coding [5] has attracted an increasing interest for DTNs [6], [7]. The benefits are increase in throughput, as well as adaptability to network topology changes and resilience to link failures. The successful reception of information does not depend anymore on receiving a specific packet, but on receiving a sufficient number of independent packets (named *Degrees of Freedom* thereafter), thereby circumventing the coupon collector problem that would emerge with single repetition of packets. In particular, [8] (chapter 10) reviews some various ways network coding can be used for opportunistic routing in DTNs, and [9] investigates deeply what benefits can NC bring in DTN under what conditions.

The goal of this paper is to design a reliable transport scheme for DTNs that aims at minimizing the round-trip file delay for a unicast session. We assume a homogeneous mobility model, time-limited epidemic routing, that the file must be split into several packets and only one packet can be exchanged during a contact, and the absence of background traffic. The scheme is based on acknowledgements and random linear coding.

Related works:
A number of reliable transport solutions for DTN have been designed in the framework of deep-space communications. Most of these solutions aim at extending the TCP mode of operation so as to handle very long delays and possible route disconnection: SCPS-TP (Space Communication Protocol Standards-Transport Protocol) [10], DS-TP (Deep-Space Transport Protocol) [11] and TP-Planet (Transport Protocol for Inter-Planetary Internet) [12] are some examples. Other solutions (e.g., Saratoga [13] or LTP-T [14]) are based on the Bundle Protocol (BP) [15] proposed by the DTN-RG [16], and are able to deliver only a hop-by-hop reliability (with

A part of the present material has been published in [1].

Lucile Sassatelli is with the Université Nice Sophia Antipolis (laboratory I3S, CNRS UMR 7271), France. Email: sassatelli@i3s.unice.fr. Arshad Ali and Manoj Panda were with Telecom SudParis, France. Arshad Ali is now with the University of Lahore, Pakistan, and Manoj Panda with the Swinburne University of Technology, Australia. Tijani Chahed is with Telecom SudParis, France. Emails: {arshad.ali,manoj.panda,tijani.chahed}@it-sudparis.eu. Eitan Altman is with INRIA Sophia Antipolis, France. Email: eitan.altman@inria.fr

This work has been partly funded by the French Research Agency under contract ANR-10-JCJC-0301.

therefore a per-hop retransmission method). Specifically, LTP-T handles errors and Automatic Repeat reQuest (ARQ) on a hop-by-hop basis. Nevertheless, all these solutions assume data delivery with probability one as soon as they know a route from the source to the destination. However, in this paper, we are interested in sparse DTN with random mobility, where assumption can be made neither about the existence of a contemporaneous path to the destination nor about knowledge of the destination location, beforehand.

For terrestrial MANETs, in order to help TCP distinguish between the losses due to congestion and the losses due to other network conditions (such as route failures), some proposals suggest that when the routing layer detects a route failure, it notifies the TCP sender about it (see [17], [4], [18] and references therein). Upon receiving this notification, the TCP sender enters a freezing state. In particular, Ad hoc TCP (ATCP) [18] utilizes network layer feedback to deal with the problem of high Bit Error Rate (BER), network congestion, and packet reordering. ATCP aims at extending TCP to MANETS, but some assumptions such as Explicit Congestion Notification (ECN)-capable node as well as sender node being always reachable are not possible to meet in a DTN context. In [19], Chen *et al.* present a solution to support TCP in disruptive MANETs by using network coding on multipath routing. The authors show that their solution is able to support high velocity and packet error rates, while achieving about 40% goodput gain over single path versions. However, this solution is designed for dense MANETs (high contact frequency) with contemporaneous routes from the source to the destination (even though they can change rapidly). These assumptions do not hold in DTNs, that are the target scenario we address in the present article.

In [20], Lucani *et al.* address the problem of the transmission over a point-to-point lossy long-delay and half-duplex channel. To minimize the expected time to completion, they determine the durations of successive cycles, during which, based on the last ACK received from the destination, the source can transmit a number of possibly redundant packets, and then stops transmitting to start to listen to the ACK. They optimize the different parameters involved in such scheme, and make use of random coding. We have been strongly inspired by this work to devise a reliable transport scheme that can optimally adapt the redundancy injected into the network, in a different context, that of DTNs, where the feature of long-delay is also characteristic. Depending on the considered buffer cleaning mechanisms, the "half-duplex" feature can appear in case of competition between source packets and acknowledgements at the nodes buffers. The lossiness may also appear due to possible packet expiry at the nodes buffers. We detail these different cases in Section III.

For DTN with random mobility, Harras and Almeroth [21] proposed different kinds of acking strategies, amongst which: a hop-by-hop approach, an "active receipt" approach and a "passive receipt" approach. The latter two provide end-to-end reliability. The active receipt is an end-to-end acknowledgement that propagates actively back to the source, i.e., as epidemically as possible. On its way back it cures the nodes that got infected by the packet corresponding to this ACK, i.e., it clears the packet from the buffers, and can prevent from infection the not-yet infected nodes too. The passive receipt is not sent back to the source in an epidemic way, but rather serves only to cure the infected nodes, that hand over the receipt only to other infected nodes. Such mechanisms are in the same spirit of those presented by Zhang *et al.* who later proposed and modeled in [22] various types of buffer cleaning, namely "Immune", "Immune-TX" and "Vaccine", the latter two corresponding respectively to the buffer cleaning strategies with "passive receipt" and "active receipt" mentioned above. The "active receipt" approach of [21] corresponds to Rule 2 in the classification provided in Section III, with $K = 1$ and buffer cleaning allowed only on ACK meeting (there is no packet expiry). In Section III, we also discuss the problem of parameter optimization under an energy constraint for this case. In this work, we consider in the same context end-to-end acknowledgements as in [21], that have the various capabilities of buffer cleaning of [22]. However, our approach is more refined than those of [21] and [22] because:

- instead of considering a single packet, we consider the general case when the file is made of K packets (as a common file transfer over TCP), thereby changing the parameters involved in the system, and hence the modeling and the optimization;
- we consider random coding to increase the efficiency of the protocol. Coding is not considered by the two above references [21] and [22];
- we analyze and provide solutions to the optimization problem (not addressed in any of the above work) of the transmission parameters, under an energy constraint, depending on the chosen buffer policy. This general problem of optimizing the routing policy while taking into account the consumed energy is of paramount importance in DTNs where the node battery lifetime is limited and dependent on the number of radio transmissions;
- in particular, for a specific important case not considered yet in the literature, we devise a reliable transport scheme that can adapt the redundancy injected into the network recursively and optimally (to mininimize the completion time under an energy constraint). To do so, we derive a fluid model of the designed system, extending the model introduced in [22] to take into account a file made of several packets, the different kinds of ACKs, the new system parameters as well as the recursive transmission protocol devised here.

Finally, in [23], Bulut *et al.* consider a multi-period spraying of a single-packet file so as to save as most energy as possible for a unicast communication in a heterogeneous mobility DTN, made of nodes socially clustered into communities. The goal is to add some more copies of a packet after certain timeout periods, in case no feedback is received from the destination to signal that the packet made it successful to it. However, the authors do not tackle the problem of acknowledgement transmission, and only assume a parallel, instantaneous and non-lossy feedback channel.

Contributions: The contributions of this paper are three-

fold:

- We design a reliable transport scheme for DTNs under opportunistic routing, using ACKnowledgements (ACKs) and coding at the source.
- We carry out the analysis of this transport scheme through mean-field approximations leading to a fluid model of the network state. Quantities of interest are then derived with coupled Ordinary Differential Equations (ODE). The so-obtained analytical model is tractable for a wide range of possible parameters, and its accuracy is validated through numerical simulations.
- We conduct a detailed study of the impact of our system's parameters on performance. Specifically, the metrics we consider are mean file completion delay (until the delivery of the last ACK at the source) and energy consumption. Based on the analytical performance model, we then devise an optimization strategy able to jointly choose the optimal parameters to minimize the file completion delay with or without an energy constraint, or minimize the energy consumption under some delay constraint.

II. Network Settings

We consider a unicast communication from a source node S to a destination node D. The network is a DTN made of $N+2$ wireless mobile nodes (S, D and N relays). Table I gathers the main parameters' notation used throughout the paper. The network is assumed to be sparse: the ratio between the coverage area of all nodes and the total area is low enough so that we neglect interference. We assume that two nodes are able to communicate when they are within reciprocal radio range, and that communications are bidirectional. For a number of mobility models (Random Walker, Random Direction, Random Waypoint), when the transmission range is small with respect to (w.r.t.) the mean inter-node distance and the velocity is high w.r.t. the network area, Groenevelt and Nain showed in [24] that the time between two consecutive node contacts is exponentially distributed. Like most previous important modeling works in DTNs (e.g., [24], [22], [25]), we make this assumption which allows to get tractable results. For a larger class of mobility scenarios where node motion can be correlated, it has been shown in [26], [27] that the inter-meeting time follows a power-law up to a point, and then exhibits an exponential decay. This model leads to hardly tractable results, as shown in [28], that is why we opt for the first model.

The meeting intensity is defined as the inverse of the mean of the exponential distribution. The mean number of contacts per time unit between a given pair of nodes (resp. a given node and any other node) is called intra-meeting (resp. inter-meeting) intensity and is denoted by β (resp. λ). The sparsity of network translates by keeping λ constant in N (in case β is constant, then the network gets denser with N). We assume that the file to be transferred needs to be split into K information packets: this occurs owing to the finite duration of contacts among mobile nodes or when the file is large with respect to the buffering capabilities of nodes. The file is considered to be well received if and only if all the K packets of the source are recovered at the destination.

Random Network Coding:
An original information packet is made of L symbols (lying in the finite field $\mathbb{F}_q$ of order q): $\mathbf{M}^i = [M_1^i, \ldots, M_L^i]$, $i = 1, \ldots, K$. The matrix of original packets is denoted by $\mathbf{M}$: $\mathbf{M} = \begin{bmatrix} \mathbf{M}^1 \\ \vdots \\ \mathbf{M}^K \end{bmatrix}$, of size $K \times L$. A coefficient vector (or "encoding vector") is made of K elements (lying in $\mathbb{F}_q$): $\mathbf{g}^j = [g_1^j, \ldots, g_K^j]$, of size $1 \times K$. The j-th encoded packet is $\mathbf{X}^j = \mathbf{g}^j\mathbf{M}$, of size $1 \times L$. The additions and multiplications are performed over the field $\mathbb{F}_q$. If the coefficients of $\mathbf{g}^j$ are picked up uniformly at random in $\mathbb{F}_q$, then $\mathbf{X}^j$ is called a Random Linear Combination (RLC). We consider this way of generating coded packets in this article, whereby the name "Random coding".
If K' coded packets are generated with K' encoding vectors stored in $\mathbf{G} = \begin{bmatrix} \mathbf{g}^1 \\ \vdots \\ \mathbf{g}^{K'} \end{bmatrix}$, of size $K' \times K$, the resulting K' encoded packets are stored in $\mathbf{X} = \begin{bmatrix} \mathbf{X}^1 \\ \vdots \\ \mathbf{X}^{K'} \end{bmatrix} = \mathbf{GM}$, of size $K' \times L$. (Possibly $\mathbf{G} = \ldots \mathbf{G}_2\mathbf{G}_1$ if successive re-encodings were allowed at intermediate nodes.) The j-th RLC in the network is made of (i) a header which is the j-th row of $\mathbf{G}$, $\mathbf{g}^j$ (each coded packet stores the coefficient that have been used in the linear combination of the original packets it stems from) and (ii) a payload made of the j-th row of $\mathbf{X}$. At the sink node at a given instant in time, the set of received packets is $[\mathbf{G}', \mathbf{X}']$, which is a subset of all the encoded packets. The linear system to solve at the sink is hence getting back $\mathbf{M}$ from $\mathbf{X}' = \mathbf{G}'\mathbf{M}$. The condition to solve this system is $\mathbf{G}'$ to be full-rank, that is: the r received coded packets have been generated with K independent coefficient vectors (more than K is not possible since the number of columns of $\mathbf{G}'$ is K). When performing a reduced-row echelon form (that is a Gauss-Jordan elimination to make appear the identity matrix) of $\mathbf{G}'$, the number of so-called Degrees of Freedom (DoFs) is the number of non-zero rows (at most K). A RLC is said to bring a new DoF if, when added to $\mathbf{G}'$, the new reduced-row echelon form of $\mathbf{G}'$ gets one more non-zero row.

So the source wants to transfer a file consisting of K information packets to D. To do so, the source generates sets of RLCs of the K information packets at certain time instants. We consider time-limited epidemic routing [22]. It is worth noting that our scheme also holds for Spray-and-Wait routing [29]. When the destination receives a new RLC, whereby decreasing the number of DoFs it still misses, it sends back an ACK to S to indicate the missing number of DoFs. On the return path, epidemic forwarding is employed for ACK dissemination. The various types of buffer management are described later. We assume that the relays have buffer capacity to store at most one packet or one ACK at any point in time. In real-world, a relay could store multiple packets and ACKs, but such assumption allows to account for the other competing flows running in the DTN and sharing the buffer space at the

Symbol	*Meaning*
Network settings	
N	total number of nodes excluding the source and the destination
β	intra-meeting intensity
λ	inter-meeting intensity ($\lambda = N\beta$)
Communication settings	
K	number of information packets the file to be received by destination D from source S is made of
ACK_i	type of ACK that D keeps sending while its missing number of DoFs to K is i
K'_i	number of RLCs that S releases when it starts a new cycle having detected that i DoFs are missing at D
$t_{K'_i}$	average time to spread K'_i RLCs from S ($t_{K'_i} = K'_i/\lambda$)
$\tau_i^{(s)}$	duration of the dissemination after $t_{K'_i}$
$\tau_i^{(w)}$	time that S waits for the most up-to-date ACK after $t_{K'_i} + \tau_i^{(s)}$
τ_i	cycle i's duration
τ_e	average of the exponential distribution of the expiry time of a packet, counted once it has been copied at a node
$X_k^{(N)}(t)$	number of nodes at time t that hold a copy of the k-th RLC sent out by S in the current cycle
$x_k(t)$	same definition as above with "fraction" replacing "number"
$Y_l^{(N)}(t)$	number of nodes at time t that hold a copy of ACK_l sent out by D in the current cycle
$y_l(t)$	same definition as above with "fraction" replacing "number"

TABLE I
MAIN NOTATION USED THROUGHOUT THE PAPER

relays. This is a simple way to account for the background traffic, and allows our system to behave independently of background traffic.

Upon receiving a RLC that increases its number of received DoFs, the destination sends back a new ACK to the source indicating the number of missing DoF: if i DoF are still missing, then ACK_i denotes the type of ACK disseminated by D. The destination keeps spreading this ACK until the number of DoFs is increased by one again. The source never sends twice the same RLC. Considering the finite field order q high enough, as soon as at least K different RLCs are collected by the destination, we assume that the rank of their encoding vector is K and hence that the destination is able to retrieve the K information packets from any set of K or more different RLCs.

III. Problem formulation

Let us first describe some possible buffer management rules. In both rules below, a RLC never replaces another RLC in a node buffer.

- Rule 1: Full contention: no packet can replace any other packet: neither ACK_i can replace ACK_j even though $i < j$ nor ACK_0 can replace any RLC.
- Rule 2: No contention: ACK_i can replace ACK_j only if $i < j$, and ACK_0 can replace any RLC.

Rule 2 can be seen as the most intuitive one in a fully-secured environment. However, if the nodes do not authenticate, then it is interesting to consider that the packet of a node cannot replace the packet of another node so as to prevent pollution attacks. This corresponds to Rule 1. In such case, packet replacement can be done only through buffer cleaning, occurring from certain timeout periods described later.

The problem we tackle here is to minimize the round-trip file delay, defined as the difference between the time the first RLC is ready to be sent out and the time the last ACK, ACK_0, is received by the source. Either this is an unconstrained problem as it is, or it is constrained in case we want to limit the number of transmissions so as to limit the energy consumption.

As the source continuously listens to the ACKs, different cases are possible to solve this optimization problem:

- For Rule 2: as the source is interested only in getting ACK_0 as soon as possible, and there is no contention between ACK_0 and any other packet, the best for the source is to send RLCs continuously until getting back ACK_0. If there is an energy constraint, then the optimization problem boils down to that solved in [30].
- For Rule 1: the spreading of the RLCs must be limited so as to let enough nodes available for the backward path of ACK_0. So the parameters to optimize would be the number of RLCs to be sent out by the source, as well as the number of copies allowed for each RLC. To control the number of disseminated copies of each RLC, a spraying counter can be used, such as that developed in Spray-and-Wait routing [29]. Here we will consider that the number of copies is limited by the time a packet is allow to spread. This does not change the following reasoning. Once the optimized number of RLCs and copies have been spread out, the source can just wait for ACK_0 to come back. However, in order to even more lower the round-trip time, it may be interesting to reset the network from time to time to free nodes from RLCs so as to speed up the backward travel of ACK_0.

In the remainder of the paper, we specifically focus on Rule 1, and because it may be interested to periodically reset the network as mentioned above, we consider a reliable transport scheme based on Rule 1 and working with cycles.

IV. The proposed reliable transport mechanism for opportunistic routing

We propose a scheme based on Rule 1 above, in which transmission is organized within cycles. During one cycle, the source sends a given number of RLCs (which is a function of the number of missing DoFs) in a back to back manner without waiting for any feedback from the destination.

At each cycle, based on the number (say i) of DoFs still missing at the destination that the source estimates by the lowest-index ACK it has received (say ACK_i), S sends

back-to-back a certain number K_i' of RLCs that it allows to propagate until time $K_i'/\lambda + \tau_i^{(s)}$. After that, the source then waits for a certain time $\tau_i^{(w)}$, collecting the ACKs which come back. The cycle ends at the expiry of $K_i'/\lambda + \tau_i^{(s)} + \tau_i^{(w)}$, where the network is reset: the buffers of all nodes are cleared. At the end of a cycle, either the lowest-index ACK received by S is ACK_0, in which case the file transmission process ends as the reception of the whole file is acknowledged, or a new cycle starts up with the new lowest-index ACK received so far. In addition to the reset of the network at the end of each cycle, we introduce an additional buffer expiry parameter for each generated RLC (be it generated by S or by a relay): when the buffer expiry time expires, the packet is dropped by the node.

A. Algorithm

Our scheme involves several timers, namely, the spreading time $\tau_i^{(s)}$, the waiting time $\tau_i^{(w)}$, the cycle timeout τ_i, each of which is a function of the number i of DoFs missing at D, from the point of view of the source indicated by the ACKs (which might be different from the actual number of received DoFs at D), and the buffer expiry τ_e. In Section IV-B, we detail how the timers can be implemented.

- **Initialization:** $i \longleftarrow K$.
- **While** $i > 0$,
  - Start of a new cycle with i missing DoFs (as viewed by the source). The source sends K_i' RLCs back to back. Each time an empty relay meets the source, the source gives a new RLC to the relay until K_i' RLCs have been sent out.
  - Each RLC is spread among the relays until $t_{K_i'} + \tau_i^{(s)}$ in an epidemic fashion, where $t_{K_i'}$ is defined as the mean time required for K_i' meetings of the source with K_i' relays: we take $t_{K_i'} = \frac{K_i'}{\lambda}$.
  - After the expiry of $t_{K_i'} + \tau_i^{(s)}$, the spreading of RLC stops (no RLC are neither released nor copied anymore), the source waits further for a duration $\tau_i^{(w)}$: the source waits for ACKs from $t_{K_i'} + \tau_i^{(s)}$ to $t_{K_i'} + \tau_i^{(s)} + \tau_i^{(w)}$. The purpose of the waiting time is to allow the ACKs to reach the source. Hence the cycle lasts for a total duration of
  $$\tau_i := t_{K_i'} + \tau_i^{(s)} + \tau_i^{(w)}.$$
  - At the end of the cycle: (i) all the relays drop the copy of the RLC or ACK they have, and (ii) the source considers the lowest-index ACK it has received so far. Let this index be j.
  - **Update:** $i \longleftarrow j$.
- **End While**.

B. Implementation details

The identifier of the connection (in the form of a source identifier and a port identifier), K, the cycle time-out τ_i and the spreading time $\tau_i^{(s)}$ are included in each RLC generated by the source. The latter two are copied in each copy and decremented at each time slot (such as each minute, assuming that each node has an internal clock) by the node they are carried by at the current time. The buffer expiry time-out τ_e is generated afresh by each relay at the time of receiving a copy of the RLC, since τ_e is local to each relay. After having been sent out by the source, each RLC spreads until time instant $t_{K_i'} + \tau_i^{(s)}$, and is dropped from the relay buffer at the earliest of the time-outs τ_i and τ_e. Since the destination generates ACKs only after receiving an RLC in the current cycle, the remainder of the cycle time-out τ_i is copied into the destination's buffer and subsequently included in the ACKs as well.

V. Analytical modeling

We now present the performance analysis of the transport scheme described above. This modeling allows to predict both file round-trip delay (up to delivery of the last ACK at the source) and energy consumption deemed as the total number of transmissions. It is later used for the optimization of the system parameters $\{(K_i', \tau_i^{(s)}, \tau_i^{(w)})\}_{i=1}^{K}$ (details are given in Section VI-C).

The modeling is based on a mean-field approximation which leads to a so-called "fluid model". In the first sub-section below, we consider the modeling of a single cycle with a direct, yet complex, model. The second sub-section presents a simpler fluid model, that allows to run it for optimization. Based on that, the third sub-section models the sequence of cycles corresponding to the complete system's mode of operation.

A. The dynamics of one cycle: a direct yet complex model

The goal is to predict the evolution over time of the different numbers of nodes describing the dissemination process and defined in Table I. To do so, we resort to a mean-field approximation that allows to predict the mean behavior of a system, modeled as a Markov chain, made of a growing number of interacting objects. The interest of a mean-field approximation lies in the fact that its complexity does not depend anymore on the number of states of the previous Markov chain, that is on the number of nodes. It therefore allows to analyze networks of any size. We give below an informal description of mean-field approximation.

The quantities $X_k^{(N)}(t)$ and $Y_l^{(N)}(t)$ defined in Table I (for k and l describing the indices of RLCs and ACKs, respectively), are random processes depending on the random mobility process. The drift of the fraction $x_k^{(N)}(t)$ (or $y_l^{(N)}(t)$), is the mean variation of this quantity between two time instants, given the state of the other quantities. Under some conditions on the drift, specifically that its limit when N tends to infinity exists (see [31], [32] for more details), Theorem 3.1 of [31] shows that the random process converges to a deterministic process when N tends to infinity. We apply this theorem to the random processes $x_k^{(N)}(t)$ and $y_l^{(N)}(t)$, because we are able to verify all the conditions listed in [32], page 7 (for the sake of simplicity, we do not develop these verifications that do not hold any subtlety). Hence, $x_k^{(N)}(t)$ and $y_l^{(N)}(t)$ converge to deterministic processes when N tends to infinity. The so-called mean-field approximation consists in considering this limit as an approximation for the

random process for finite N. The deterministic limit processes of interest are the solutions of (possibly coupled) Ordinary Differential Equations (ODEs). This finite limits expressed with ODEs are called the fluid model. As these limit processes do not depend on N anymore, we keep the notation of the random process but the superscript (N). The limits are limits of fractions of nodes, the approximation of the number of nodes for a given N is deduced from the respective fraction, that is, the deterministic fluid approximation for the processes $\{X_k^{(N)}(t)\}_{t\geq 0}$ and $\{Y_l^{(N)}(t)\}_{t\geq 0}$ is given by

$$X_k^{(N)}(t) \approx N x_k(t) \quad , \quad Y_l^{(N)}(t) \approx N y_l(t) \; . \tag{1}$$

Let us now present the ODEs of the fluid model.

We denote a cycle that starts up with i missing DoFs by an i-cycle. We model the dynamics of the network during an i-cycle by expressing the spreading of the K_i' RLCs and the i ACKs, $ACK_0, \ldots, ACK_{i-1}$. Such modeling can be done independently of the earlier cycles, since the relays drop the copy of the RLC or ACK they have at the end of each cycle. Thus, we reset the time variable t to zero at the beginning of each cycle and study the network dynamics in an i-cycle for $t \in [0, \tau_i]$.

Let us now focus on getting these right-hand sides of the ODE approaching the network dynamics. These ODEs can be easily understood by noting that $\frac{dx(t)}{dt} = \frac{x(t+dt)-x(t)}{dt}$, so it amounts to express the infinitesimal variation of x between t and $t+dt$. This variation can be easily expressed knowing the system's mode of operation. In the ODEs, s and d denote the fraction of nodes that are the sources, and those that are the destinations, respectively. As in our system of interest there is a single source and a single destination, s and d are both set to $1/N$, where N is the total number of nodes of the considered network (that we model). Let S_l denote the random time at which the number of missing DoF at the destination changes from $l+1$ to l, $\forall l = 0, \ldots, i-1$, with the convention that $S_i = 0$. Note that $[S_{l+1}, S_l]$ denotes the time interval during which the destination remains at state $l+1$, i.e., with $l+1$ missing DoFs. In order to obtain the spreading dynamics of the ACKs, we first condition on S_l, $\forall l$, $l = 0, 1, \ldots, i-1$. We then solve the dynamic equations for ACKs together with that of the RLCs. Finally, we uncondition over all possible values of the random variable S_l, $l = 0, 1, \ldots, i-1$ using their joint density function.

$$\frac{dx_k(t)}{dt} = \begin{cases} \lambda \left(s\, pois_k(t) + x_k(t)\right)\left(1 - x(t) - y(t)\right) - \\ d\lambda x_k(t) - \mu x_k(t), \text{ for } 0 < t \leq \tau_{K_i'} + \tau_i^{(s)}, \\ -d\lambda x_k(t) - \mu x_k(t), \text{ for } \tau_{K_i'} + \tau_i^{(s)} < t \leq \tau_i. \end{cases} \tag{2}$$

$$\frac{dy_l(t)}{dt} = \begin{cases} 0, \text{ for } 0 \leq t < S_l, \\ \lambda\left(d + y_l(t)\right)\left(1 - x(t) - y(t)\right) + d\lambda x(t) - \\ s\lambda y_l(t), \text{ for } \{S_l \leq t < S_{l-1} \text{ and } l > 0\} \\ \text{ or } \{S_l \leq t < \tau_i \text{ and } l = 0\} \\ \lambda y_l(t)\left(1 - x(t) - y_l(t)\right) - s\lambda y_l(t), \text{ for} \\ \{S_{l-1} \leq t \leq \tau_i \text{ and } l > 0\} \end{cases} \tag{3}$$

with initial conditions, $x_k(0) = 0$, $\forall k = 1, 2, \ldots, K_i'$ and $y_l(0) = 0$, $\forall l = 0, 1, \ldots, i-1$, where $\mu = 1/\tau_e$, $x(t) := \sum_{k=1}^{K_i'} x_k(t)$, $y(t) := \sum_{l=0}^{i-1} y_l(t)$ and $pois_k(t)$ denotes the probability that the source has met exactly $k-1$ relays up to time t. From the Poisson distribution of the number of meetings per unit of time (stemming from the exponential distribution of inter-meeting times), we have:

$$pois_k(t) = \exp^{-\lambda t} \frac{(\lambda t)^{k-1}}{(k-1)!} \; .$$

In equations (2) and (3), the quantities $x_k(t)$ and $y_l(t)$ represent the mean fraction of relays that have a copy of RLC k and ACK l, respectively. Hence, the quantity $x(t)$ (resp. $y(t)$) denotes the mean fraction of relays that have a copy of any RLC (resp. any ACK). The term $s\lambda p_k(t)(1-x(t)-y(t))$ corresponds to the contribution of the source to the increase of $x_k(t)$ per unit of time, while $\lambda x_k(t)(1-x(t)-y(t))$ (resp. $\lambda y_l(t)(1-x(t)-y(t))$) is the contribution from the yet-infected relays. The term $d\lambda x_k(t)$ corresponds to the replacement of a copy of RLC k by an ACK at the destination. The term $\mu x_k(t)$ corresponds to the RLC drop at relays due to buffer expiry time-out. We have $\mu = 1/\tau_e$ that represents the probability that a RLC expires within a unit of time.

In order to be able to compute the mean completion time, we have to express the $P_{X_k}(t)$'s and the $P_{Y_l}(t)$'s which denote the probability that the destination and source have received a copy RLC k and ACK l, respectively, by time t. We denote the Cumulative Distribution Function (CDF) associated with the delay D_l of ACK l, conditioned upon $S_{i-1}, S_{i-2}, \ldots, S_0$, by $P_{Y_l}(t, s_{i-1}, s_{i-2}, \ldots, s_0) := Pr\{D_l \leq t | S_{i-1} = s_{i-1}, \ldots, S_0 = s_0\}$. They are given by

$$\begin{aligned} \frac{dP_{X_k}(t|s_{i-1},\ldots,s_0)}{dt} &= \lambda x_k(t)(1 - P_{X_k}(t|s_{i-1},\ldots,s_0)) \\ \frac{dP_{Y_l}(t|s_{i-1},\ldots,s_0)}{dt} &= \lambda y_l(t)(1 - P_{Y_l}(t|s_{i-1},\ldots,s_0)) \end{aligned} \tag{4}$$

with initial conditions $P_{X_k}(0) = 0$ and $P_{Y_l}(0, s_{i-1}, s_{i-2}, \ldots, s_0) = 0$. The CDF $P_{Y_l}(t)$ is obtained by un-conditioning as

$$P_{Y_l(t)} = \int_{s_{i-1}=0}^{t} \int_{s_{i-2}=s_{i-1}}^{t} \cdots \int_{s_0=s_1}^{t} f_{S_{i-1}\ldots S_0}(s_{i-1}, \ldots, s_0) \; P_{Y_l}(t|s_{i-1}, \ldots, s_0) ds_{i-1}, \ldots ds_0 \tag{5}$$

where $f_{S_{i-1}\ldots S_0}(s_{i-1}, \ldots, s_0)$ denotes the joint density of the random variables S_l, $\forall l$, $l = 0, 1, \ldots, i-1$, which we derive now below.

Let E be a set of pairwise different elements from set $T = \{1, ..., K_i'\}$, whose cardinal is $|E|$, and $\mathbf{v}$ be a vector storing the occurrence probabilities of each element of T. We define $R(K_i', z, \mathbf{v})$ as the probability that exactly z elements of T occur, given the occurrence probability vector $\mathbf{v}$. We have, assuming independence of elements in T:

$$R(K_i', z, \mathbf{v}) = \sum_{E \in T : |E| = z} \prod_{i \in E} v_i \prod_{i \in T \setminus E} (1 - v_i) \; .$$

In [33], a fast method based on dynamic programming and Fast Fourier Transform (FFT) is presented to lower the complexity of the calculation of the $R(.,.,.)$ function. This allows us to use it in practice to tackle reasonable numbers K_i' of RLCs (from several tens up to a few hundreds). As the time S_m where ACK_m starts to be spread out by the destination does not depend on S_n for $n < m$, we can express $f_{S_{i-1}\ldots S_0}(s_{i-1}, \ldots, s_0)$ as

$$f_{S_{i-1}\ldots S_0}(s_{i-1}, \ldots, s_0) = \prod_{m=0}^{i-1} f_{S_m | s_{i-1}\ldots s_{m+1}}(s_m)$$

with

$$f_{S_m|s_{i-1}\dots s_{m+1}}(s_m) = \frac{dR(K_i', m, \mathbf{P_X}(s|s_{i-1},\dots,s_{m+1}))}{ds}$$

and $\mathbf{P_X}(s|s_{i-1},\dots,s_{m+1}) = [P_{X_1}(s|s_{i-1},\dots,s_{m+1}),\dots,P_{X_{K_i'}}(s|s_{i-1},\dots,s_{m+1})]$. Note that $P_{X_k}(s|s_{i-1},\dots,s_{m+1})$ is the same as $P_{X_k}(s|s_{i-1},\dots,s_0)$, $\forall s_m,\dots,s_0$ and $\forall k = 1,\dots,K_i'$. Note that the components of $\mathbf{P_X}(s|s_{i-1},\dots,s_{m+1})$ are independent because they stem from the above set of ODEs that model the mutual dependence of every kind of packets.

B. The dynamics of one cycle: a simpler fluid model

Owing to the multiple time integrals involved in equation (5) and needed to obtain the delay CDF of each ACK, the numerical implementation becomes quickly too computationally intensive as the number K of information packets increases. However, the results of Benaïm and Le Boudec [32] can be applied to the network system we have, allowing to model the dependence of the spreading process on the destination state equivalently to equation (5) but without explicit probability conditioning. Specifically, the network is made of N objects (relay nodes) having states $S_n^N(t)$, $n = 1,\dots,N$, in the finite set $\mathcal{S} = \{0,1,\dots,K_i'+i\}$ (the node's buffer is empty, or has RLC_k, $k = 1,\dots,K_i'$, or has ACK_l, $l = 0,\dots,i-1$). The destination is modeled as a finite resource with state $R^N(t)$ in $\mathcal{R} = \{0,\dots,i-1\}$ (the state is the number of DoFs still missing at D). The process $(A_1^N(t),\dots,A_N^N(t),R^N(t))$ is a homogeneous Markov chain for which it can be easily shown it is a mean field interaction process [32]. Then from Theorem 1 of [32], we can state that, under the conditions mentioned in the second paragraph of this section, when N increases, the fraction of nodes infected by RLC_k or ACK_l, for $k = 1,\dots,K_i'$ and $l = 0,\dots,i-1$, converge to deterministic processes that are the solutions of, respectively:

$$\frac{dx_k(t)}{dt} = \sum_{j=0}^{i-1} p\left(j, x_0(t),\dots,x_{K_i'}(t), y_0(t),\dots,y_{i-1}(t)\right) f_{x_k}\left(j, x_0(t),\dots,x_{K_i'}(t), y_0(t),\dots,y_{i-1}(t)\right)$$

$$\frac{dy_l(t)}{dt} = \sum_{j=0}^{i-1} p\left(j, x_0(t),\dots,x_{K_i'}(t), y_0(t),\dots,y_{i-1}(t)\right) f_{y_l}\left(j, x_0(t),\dots,x_{K_i'}(t), y_0(t),\dots,y_{i-1}(t)\right)$$

where $p\left(j, x_0(t),\dots,x_{K_i'}(t), y_0(t),\dots,y_{i-1}(t)\right)$ denotes the probability, which keeps constant with N, that the resource (i.e., the destination) is in state j given the fraction of nodes in each state, and $f_{x_k}\left(j, x_0(t),\dots,x_{K_i'}(t), y_0(t),\dots,y_{i-1}(t)\right)$ (resp. $f_{y_l}(.)$) denotes the limit when N tends to infinity of the mean increase in the fraction of nodes infected by RLC_k (resp. ACK_l), given the fraction of nodes in each state.

Thus we obtain a simpler model as compared to that presented above, that we shall therefore use when computing the model, especially in the parameter optimization step presented in Section VI-C.

$$\frac{dx_k(t)}{dt} = \begin{cases} \lambda\left(s\ pois_k(t) + x_k(t)\right)\left(1 - x(t) - y(t)\right) - \\ \lambda d x_k(t) - \mu x_k(t), \text{ for } 0 < t \le \tau_{K_i'} + \tau_i^{(s)}, \\ -\lambda d x_k(t) - \mu x_k(t), \text{ for } \tau_{K_i'} + \tau_i^{(s)} < t \le \tau_i. \end{cases} \quad (6)$$

$$\frac{dy_l(t)}{dt} = \begin{cases} \lambda y_l(t)\left(1 - x(t) - y(t)\right)\sum_{m=0}^{l} q_m(t) + \\ \lambda d\left(1 - y_l(t)\right) q_l(t) - \lambda s y_l(t)\sum_{m=0}^{l} q_m(t) - \\ \lambda d y_l(t)\sum_{m=0}^{l-1} q_m(t), \text{ for } l > 0 \\ \lambda y_l(t)\left(1 - x(t) - y(t)\right)\sum_{m=0}^{l} q_m(t) + \\ \lambda d\left(1 - y_l(t)\right) q_l(t), \text{ for } l = 0 \end{cases} \quad (7)$$

with the same initial conditions and definitions as in the previous model. The ODE for $x_k(t)$, $\forall k = 1,\dots,K_i'$ has not changed, and the ODE for $y_l(t)$ is obtained in the same way as for equation (3), for all $l = 0,\dots,i-1$, accounting additionally for the state of the destination in each term: $q_m(t)$, for $m = 0,\dots,i-1$, denotes the probability that exactly $i-m$ DoFs have been received at a destination (i.e., $i-m$ different RLCs), whereby $q_m(t) = Sz(K_i', i-m, \mathbf{P_X}(t))$. When the destinations spread ACK_l at time t, they cannot give such ACK only to the nodes that yet have ACK_l. Finally, $-\lambda s y_l(t)$ is the amount of node's buffers carrying an ACK that are cleared upon meeting with the source and $-\lambda y_l(t)\sum_{m=0}^{l-1} q_m(t)$ is the amount of buffers holding ACK_l that are overwritten by the updated ACK upon meeting with the destination.

The CDF of the delays of RLCs and ACKs are then expressed in the same way as in the previous section:

$$\frac{dP_{X_k}(t)}{dt} = \lambda x_k(t)(1 - P_{X_k}(t))$$
$$\frac{dP_{Y_l}(t)}{dt} = \lambda y_l(t)(1 - P_{Y_l}(t))$$

C. Modeling the sequence of cycles

Let us now finalize the modeling of the proposed transport scheme by accounting for the different possible successions of cycles so as to get the expected file completion time. We recall that the file transfer has been successful when the source receives ACK_0, indicating that the destination could recover the whole file. Let us first express the transition probabilities that govern the state transition of the source: $P_{ij}(t)$ is the probability that the lowest-index ACK received by the source at t is j, given that the source starts up with a i-cycle. Thus, $P_{ij}(\tau_i)$ is the probability that the source state at the end of cycle i be j (i.e., S starts up with a j-cycle if $j > 0$). Remember that $P_{Y_l}(\tau_i)$, for $l = 0,\dots,i-1$ denotes the probability that the source has received ACK l by the end of the i-cycle. Hence the transition probabilities $P_{ij}(t)$, $j = 0,\dots,i$, are given by

$$P_{ij}(t) = P_{Y_j}(t)\prod_{l=0}^{j-1}(1 - P_{Y_l}(t))\ ,\ \text{for } j \neq i,$$

$$\text{and } P_{ii}(t) = 1 - \sum_{j=0}^{i-1} P_{ij}(t)\ .$$

Let T_i, $i = 1,\dots,K$, denote the expected time to reach the source state 0 starting from the beginning of an i-cycle. Hence, T_K represents the expected delay for the source to receive

ACK_0. Let $E_{Direct}[T_{i\to 0}]$ denote the expected time to reach source state 0 by the end of the current i-cycle, given that state 0 is reached by the end of the current i-cycle. Then, since a j-cycle with $j < i$ is entered only after the end of the current i-cycle, hence waiting for τ_i, we have:

$$T_i = P_{i0}(\tau_i)E_{Direct}[T_{i\to 0}] + (1 - P_{i0}(\tau_i))\tau_i + \sum_{j=1}^{i} P_{ij}(\tau_i)T_j \ . \tag{8}$$

Let us now express $E_{Direct}[T_{i\to 0}]$. Let $f_{T_{i\to 0}}(t)$ and $F_{T_{i\to 0}}(t)$ denote the Probability Distribution Function (PDF) and CDF, respectively, of ACK_0 delay given that this delay is lower than τ_i. The definition is $E_{Direct}[T_{i\to 0}] = \int_0^{\tau_i} t f_{T_{i\to 0}}(t)dt$. An integration by parts gives $E_{Direct}[T_{i\to 0}] = \tau_i F_{T_{i\to 0}}(\tau_i) - \int_0^{\tau_i} F_{T_{i\to 0}}(t)dt$. The conditioning on the event that the delay is lower than τ_i gives $F_{T_{i\to 0}}(t) = \frac{P_{i0}(t)}{P_{i0}(\tau_i)}$. Whereby, substituting the final expression of $E_{Direct}[T_{i\to 0}]$ into equation (8) leads to:

$$T_i = \frac{\sum_{j=1}^{i-1} P_{ij}(\tau_i)T_j + \tau_i - \int_0^{\tau_i} P_{i0}(t)dt}{1 - P_{ii}(\tau_i)}. \tag{9}$$

VI. Numerical assessments and impact of the parameters

A. Accuracy of the model and impact on delay

In this section, we assess numerically the fit between the above analytical model and the simulations, and we study the impact of each parameter on the system performance.

The simulations have been carried out with Matlab using a discrete event simulator, where the inter-meeting time between each pair of nodes is generated with an exponentially distributed random variable with mean $1/\beta$, so is generated with mean τ_e the expiry time from the instant a buffer receives a packet. Each experiment has been averaged over 200 independent runs, and the 95% confidence interval are plotted in the figures. The metric we first use is the mean delay for the source to receive ACK_0, referred to as the "mean file completion delay" hereafter. We set the number of nodes N, β, K fixed. For the sake of simplicity of the analysis, we restrict the values of $\tau_i^{(s)}$ and $\tau_i^{(w)}$ to be constant in i, i.e., $\tau_i^{(s)} = \tau_s$ and $\tau_i^{(w)} = \tau_w$ over all cycles $i = 1, \ldots, K$. We proceed in the same way for K'_i except that $K'_i = i + K'$ for some fixed K', for all $i = 1, \ldots, K$. The parameters we vary are K', τ_s and τ_w and τ_e. For the analysis of the impact of one parameter, we set the other three fixed. In all experiments, we set $N = 100$, $\beta = 0.005$ and $K = 10$.

In Figures 1, 2 and 3, we can see the evolution of the mean file completion delay against different parameters in various configurations, both from simulations and analytical (fluid) model derived in Section V.

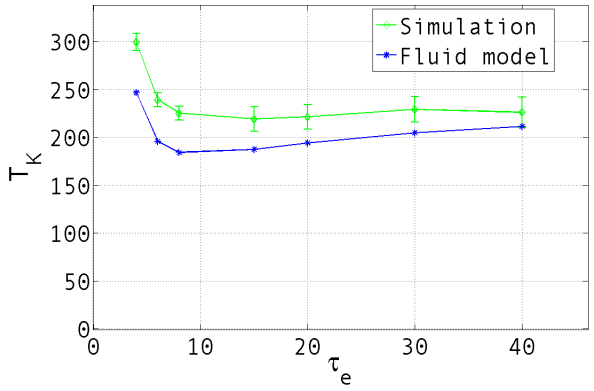

Fig. 1. Mean completion delay against τ_e, $N = 100$, $\beta = 0.005$, $K = 10$, $K' = 5$, $\tau_s = \tau_w = 10$.

Let us first analyze the impact of the buffer expiry time τ_e. We set $\tau_s = \tau_w = 10$ and $K' = 5$.

We first discuss the accuracy of our fluid model, in particular the impact of τ_e on the accuracy. We can see from Figure 1 that the accuracy of our analytical model is good for high values of τ_e but worse for lower values (around 10). Figure 2 represents the mean completion delay against each of the parameters K', τ_s or τ_w. Figures 2.a, 2.b and 2.c in the left column are for $\tau_e = 10$, while those in the right column (2.d, 2.e and 2.f) are for $\tau_e = 100$. We can see that the fit between the analytical model and simulations improves when τ_e increases. Moreover, for a given τ_e, either high (100) or low (10), the fit is better for lower values of K', τ_s or τ_w. We identified that the inaccuracy in modeling τ_e impacts the number of cycles estimated by the model (it estimates a too low number of cycles). Hence, the higher the duration of a cycle, the bigger the inaccuracy in T_K. That is why we observe less accuracy with higher values of K', τ_s or τ_w (as the cycle duration is an increasing function of these values). However, it is worth noting that, despite this relative gap between the analytical model and the simulations, having analyzed the root cause of this gap allows to be sure that the trends are similar. This allows us to assume that our model will lead to consistent optimization of the system parameters (discussed in subsection VI-C). Note that the modeling of the expiry time is the same as in [22].

Now we discuss the impact of τ_e on the mean completion delay T_K. Fig. 1 shows that there is a minimum of the completion delay in τ_e. When τ_e is very low, the delay is high as the packets expire too early to allow dissemination of the RLCs. As τ_e increases, this effect vanishes and the delay decreases. When τ_e is at the same time high enough to enable sufficient dissemination of the RLCs and low enough to help cleaning the buffers to make some room available for the ACKs to come back, the delay is minimum. However, when τ_e is too high, the number of disseminated RLCs is high but the buffers are not cleaned, so the ACK cannot easily come back to the source, whereby an increase in the delay.

Let us then analyze the impact of K'. In Fig. 2.a, we set $\tau_s = \tau_w = \tau_e = 10$. We can see that there is a minimum

of the expected file completion time in K'. Indeed, if no redundancy is added (corresponding to $K' = 0$), numerous copies of the same packet will occupy the network, whereby an increase of the one-way delay because the destination will need more time to collect K different RLCs. When redundancy is added, i.e., more RLCs than the requested number of DoFs are disseminated by the source (remember that the source never sends twice the same RLC), then the performance increases as the destination is able to grab the needed number of RLCs more rapidly (equivalently, with higher probability in a given time). However, as the amount of redundancy sent out increases, more relays get occupied and the ACKs have less room to disseminate, and hence the completion delay increases.

We now focus on the impacts of τ_s and τ_w. Figures 2.b to 2.c represent the impact of these parameters depending on the level of redundancy K'.

- When no redundancy is added ($K' = 0$), the delay has a minimum in τ_s (see Fig. 2.b). Indeed, τ_s is the time the RLCs are allowed to spread once all of them have been sent out by the source. When the source does not spread more RLCs than the number of required DoFs, then the spreading time τ_s helps relays to get a copy of a RLC. Thus, the delay first decreases when τ_s increases. However, if τ_s increases too much, the same issue as with K' above arises: the network gets congested owing to too many copies of RLCs occupying relays' buffers, and the ACK cannot come back rapidly to the source (because the cycle duration, i.e., the reset period of the network, increases with τ_s). The impact of τ_w (see Fig. 2.c) requires another interpretation. We also observe a minimum of the delay in τ_w. However, τ_w does not control the RLC spreading, but the time the source waits for an ACK, i.e., the cycle duration that is equivalent to a reset period of the network: when τ_w is too low, the buffers are cleaned from RLCs and ACKs (end of cycle) before the last ACK could reach the source, but if τ_w gets too high then it does not help anymore at the right time to clean the relays' buffers so as to allow them to carry copies of ACKs to the source in some cycle subsequent to the cycle where RLCs have been received at the destination.
- When a lot of redundancy is added ($K' = 20$), then τ_s and τ_w are detrimental to performance (see Fig. 3.a and 3.b): the only time of RLCs spreading by the source is sufficient to get a high diversity of RLCs and number of copies of RLCs, and to let the final ACK to come back. That is, once all the RLCs are sent out, if they are enough, it is very likely that the destination grabbed K different RLCs within $t_{K'_K}$. Therefore, it is good to allow the network to reset immediately after $t_{K'_K}$ to clean the relays' buffers and speed up the ACK back trip. This result is to be put in light with the experiment in Fig. 2.a where K' too high is shown to be detrimental. However, it must be noticed that the experiment of Fig. 2.a is for some fixed non-negligible $\tau_s = 10$ and $\tau_w = 10$: if the network reset occurs much after the dissemination of all RLCs, then the final ACK which will try to come back in the first cycle, will not be able to meet enough free relays and will have to wait until the end of the cycle to have free relays to be able to reach the destination.

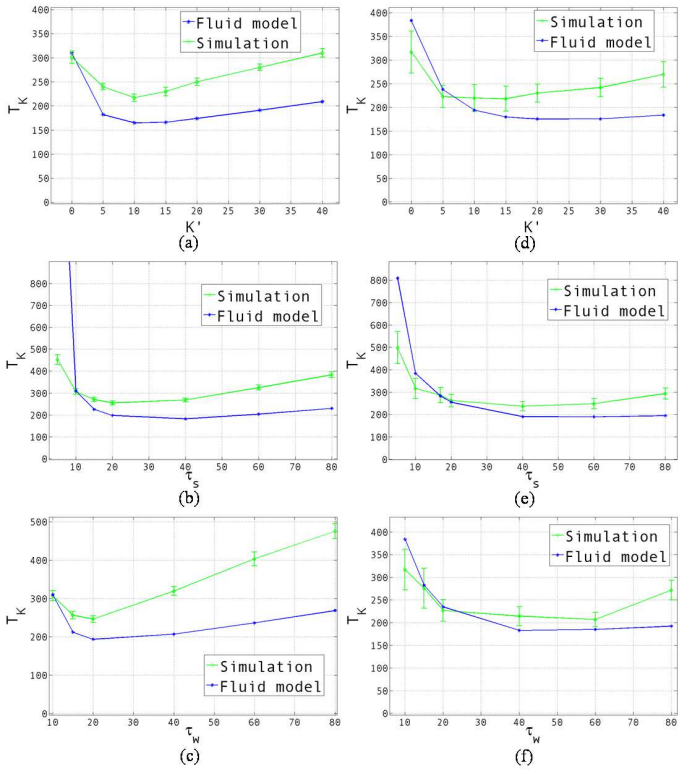

Fig. 2. $N = 100$, $\beta = 0.005$, $K = 10$. Mean completion times against K', τ_s or τ_w. The values of the parameters other than that on the x-axis are $K' = 0$, $\tau_s = \tau_w = 10$. The left column (a, b, c) is with $\tau_e = 10$, while the right column (d, e, f) is with $\tau_e = 100$.

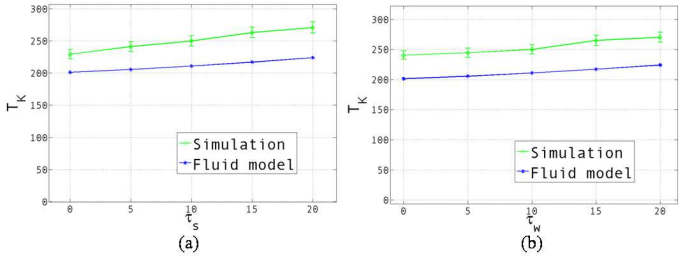

Fig. 3. $N = 100$, $\beta = 0.005$, $K = 10$, $K' = 20$, $\tau_e = 10$. (a) Mean completion delay against τ_s. (b) Mean completion delay against τ_w.

B. Impact on energy consumption

Another metric of interest in DTN is energy consumption: as nodes are mobile, one must keep in mind to preserve battery lifetime while trying to get as good performance (in terms of throughput or delay) as possible. We investigate the impact of our system parameters on the achievable trade-off between energy and performance. In the next subsection, we formally express and discuss the general optimization procedure. Let us first express the energy consumption. We define it as the product of the unitary cost of a transmission and the mean total number of RLC transmissions that occurred in the network up to delivery of ACK_0 at the source, and we denote the latter by C_K (if the file to be delivered is made of K packets). Slightly abusing the notation, we refer to C_K as the energy consumption. Then the derivation of C_K is similar to that of

T_K in Eqn. 9, and we get, when considering that the expiry time is very large:

$$C_K = \frac{\sum_{j=1}^{i-1} P_{ij}(\tau_i)C_j + P_{i0}(\tau_i)x(T_{dir}) + (1 - P_{i0}(\tau_i))x(\tau_i)}{1 - P_{ii}(\tau_i)} ;$$

with $T_{dir} = E_{Direct}[T_{i\to 0}]$ expressed in the previous section. For low expiry time, this equation must be slightly modified so as to account for transmissions of packets that expired at the time of completion (only the term $x(T_{dir})$ has to be modified). This amounts to a straightforward extension that we do not discuss for the sake of simplicity. That is why in Fig. 4, we consider $\tau_e = 1000$.

Fig. 4 represents both T_K and C_K, for $K = 10$, against K' for two different fixed values of τ_w in 4.a, and against τ_w for two different fixed values of K' in 4.b. In Fig. 4.a, when K' increases, C_K decreases for both values of τ_w (0 and 20). The reason for this is that when K' increases, the cycle duration also increases, hence the network reset occurs later than with a lower K'. Thus, the file completion delay gets higher because the relays are not freed early enough, but as less cycles are needed, energy still decreases. As well, in Fig. 4.b, when τ_w increases, the probability that completion occurs in the first cycle is higher because the cycle duration τ_i increases, hence energy decreases as it is less likely that other cycles (with other bursts of RLCs transmissions) are needed, although the mean delay increases. Also, we observe in Fig. 4.a that when K' is low, energy consumption is higher for $\tau_w = 0$. Indeed in such case, the mean completion delay T_K is higher, and hence energy consumption up to completion is higher (a higher number of cycles are required in average).

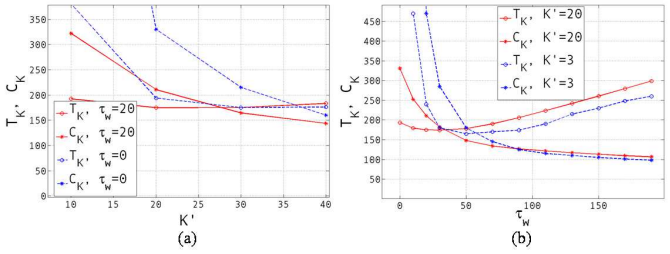

Fig. 4. $N = 100$, $\beta = 0.005$, $K = 10$, $\tau_e = 1000$ and $\tau_s = 0$. (a) Mean completion delay T_K and number of transmission C_K against K'. (b) Mean completion delay T_K and number of transmission C_K against τ_w.

C. General optimization procedure

Let us now discuss a strategy to select the optimal parameters (i.e., the triples $(K'_i, \tau_i^{(s)}, \tau_i^{(w)})$ for $i = 1 \ldots, K$). The objective we consider is the minimization of the expected file completion time, i.e., T_K defined above, under an energy consumption constraint (maximum number of transmissions). The problem of finding the optimal parameters $\{(K'_i, \tau_i^{(s)}, \tau_i^{(w)})\}_{i=1}^{K}$ (which control the dissemination of RLCs and ACKs), under some energy constraint can be formulated as:

$$\min_{\{(K'_i, \tau_i^{(s)}, \tau_i^{(w)})\}_{i=1}^{K}} T_K$$

Lucani *et al.* pointed it out in [20] that the optimization of T_K can be recursive starting with optimizing T_1, then T_2 and so on, owing to the expression of T_K in equation (9):

$$\min(T_i) = \frac{\sum_{j=1}^{i-1} P_{ij}(\tau_i)\min(T_j) + \tau_i - \int_0^{\tau_i} P_{i0}(t)dt}{1 - P_{ii}(\tau_i)}.$$

It is worth noting that such sequential optimization can be conducted when no energy constraint is enforced (remember that in the full contention case we consider, i.e., Rule 1 of Section III, optimization is required even without energy constraint). However, with an energy constraint, as the total energy expenditure must be computed over all the cycles, such sequential optimization cannot be used, and then it would be the $3K$-uplet that the optimization algorithm would have to jointly identify. Without energy constraint, for each optimization step i, aiming at minimizing T_i, a three-dimensional optimization must be performed so as to get the optimal triple $(K'_i, \tau_i^{(s)}, \tau_i^{(w)})$.

Be it with or without energy constraint, the optimization pertains to the class of nonlinear optimization problems. Many general algorithms for solving such problems have been developed. For example, the Differential Evolution (DE) [34] can be used. DE is a robust optimizer for multivariate functions. We do not describe DE here, but only say that this algorithm is in part a hill climbing algorithm and in part a genetic algorithm.

The above optimization procedure does not have to be performed at the source node, but is rather performed offline, and the resulting optimal parameters for each possible i-cycle, $i = 1, \ldots, K$, are stored in memory at the source node, so as to be used as needed. Note that our optimization procedure requires the knowledge of β and N that can be estimated by using the history of node meetings.

VII. Discussion

A. Considering a buffer size of one packet

Taking into account the multi-session case (or equivalently the background traffic) in the modeling of dissemination in DTNs is difficult, as shown in [35], [36]. Our purpose in this article is to design a reliable transport protocol and carry out an as exhaustive as possible analysis of the impact of parameters on the protocol performance. To do so, we therefore consider the single-session case. The multi-session case is out of the scope of this paper and left for future work. We assume a fixed-fraction of the buffer size is dedicated to the session of interest to understand the impact of the parameters in a simple setting. The qualitative results presented in this paper are not sensitive to the specific fraction of the buffer size dedicated to the session of interest, provided that the ratios with the other parameters (N, λ, K, K_i', $\tau_i^{(s)}$ and $\tau_i^{(w)}$) are preserved in an order sense. In particular, if the buffer size dedicated to the session of interest is greater than one packet, the model gets more complicated but can be adapted, in the same way as in [35]. Then the same analysis of the parameters can be carried out.

B. Impact of background traffic on the protocol performance

If other connections are running in the network and they are sharing the buffers without any reservation, the room available

in a buffer for a given session is dependent on the number of packets that the sources send out. The number of packets that a source sends out depends in turn on the number of ACKs it has received from the destination. Unlike TCP, as long as a source misses ACKs, it does not decrease the number of sent packets per cycle, if this is decided with the model which does not take into account any background traffic. Hence, the connections in their beginning are going to grab more room than those close to their completion. These connections are therefore slowed down by the new-coming ones. This is comparable to the slow-down of short flows (or close to completion) by long flows in wireline networks when FIFO is used as queue management (as opposed to size-based scheduling). Hence, if the multi-session case were considered using, e.g., the method of [35], for a given traffic profile (that is for a given distribution of flow size), we could resort to a joint optimization of the protocol parameters of each session to maximize a certain utility (such as $\sum_i U(T_i)$), T_i being the mean completion time of connexion i. Then, another important step is to design a decentralized version.

VIII. Conclusion

In this article, we have devised a reliable transport scheme for homogeneous-mobility DTN operating with opportunistic routing algorithms. This mechanism relies on ACK and coding at the source. The different versions of the problem depending on buffer management policies have been formulated, and a fluid model based on mean-field approximation has been derived for the designed reliable transport mechanism. This model allowed to express both the mean file completion time and the energy consumption up to delivery of the last ACK at the source. The accuracy of this model has been assessed through numerical simulations, that also served for investigating the impact of the system parameters on the performance. We finally presented a joint optimization of the mean completion delay with and without an energy constraint, so as to identify the optimal set of parameters to use.

Future work consists (i) in progressing to a more complete TCP counterpart for DTN by integrating rate control, e.g., building on back-pressure algorithms, and (ii) in extending this scheme to heterogeneous-mobility models.

References

[1] A. Ali, E. Altman, T. Chahed, M. Panda, and L. Sassatelli, "A new reliable transport scheme in delay tolerant networks based on acknowledgments and random linear coding," in *IEEE Int. Teletraffic Congress (ITC)*, San Francisco, USA, Sep. 2011, pp. 214–221.

[2] NSF, "NSF workshop on future directions in wireless networking, final report," Arlington, USA, Nov. 2013. [Online]. Available: http://ecedha.org/docs/nsf-nets/final-report.pdf?sfvrsn=0

[3] M. Gerla, R. Bagrodia, L. Zhang, K. Tang, and L. Wang, "TCP over wireless multi-hop protocols: simulation and experiments," in *IEEE Int. Conf. on Comm. (ICC)*, Vancouver, Canada, Jun. 1999, pp. 1089–1094.

[4] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," *ACM Wireless Networks*, vol. 8, no. 2, pp. 275–288, Mar. 2002.

[5] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE Int. Symp. on Info. Theory (ISIT)*, Yokohama, Japan, Jul. 2003, p. 442.

[6] J. Widmer and J.-Y. Le Boudec, "Network coding for efficient communication in extreme networks," in *ACM SIGCOMM workshop on Delay-tolerant networking*, Philadelphia, USA, 2005, pp. 284–291.

[7] Y. Lin, B. Li, and B. Liang, "Stochastic analysis of network coding in epidemic routing," *IEEE Journal on Selected Areas in Comm. (JSAC)*, vol. 26, no. 5, pp. 794–808, Jun. 2008.

[8] X. Zhang, G. Neglia, and J. Kurose, "Network Coding in Disruption Tolerant Networks," in *Network coding: fundamentals and applications*, M. Médard and A. Sprintson, Eds. Academic Press, 2011.

[9] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang, "Benefits of Network Coding for Unicast Application in Disruption-Tolerant Networks," *IEEE/ACM Trans. on Netw.*, vol. 21, no. 5, pp. 1407–1420, Oct 2013.

[10] Consultative Committee for Space Data Systems, "Space Communications Protocol Standards (SCPS) - Transport Protocol (SCPS-TP)," *CCSDS 714.0-B-2, Blue Book*, Oct. 2006.

[11] I. Psaras, G. Papastergiou, V. Tsaoussidis, and N. Peccia, "DS-TP: Deep-Space Transport Protocol," in *IEEE Aerospace Conf.*, 2008, pp. 1–13.

[12] O. B. Akan, J. Fang, and I. F. Akyildiz, "TP-Planet: A Reliable Transport Protocol for InterPlanetary Internet," *IEEE Journal on Selected Areas in Comm. (JSAC)*, vol. 22, no. 2, pp. 348–361, 2004.

[13] L. Wood, J. McKim, W. Eddy, W. Ivancic, and C. Jackson, "Saratoga: A Scalable File Transfer Protocol," in *IETF draft draft-woodtsvwg-saratoga-03*, Oct. 2007.

[14] S. Farrell and V. Cahill, "LTP-T: a Generic Delay Tolerant Transport Protocol," in *Technical report*, 2005. [Online]. Available: http://www.scss.tcd.ie/publications/tech-reports/reports.05/TCD-CS-2005-69.pdf

[15] K. Scott and S. Burleigh, "Bundle Protocol Specification," in *IETF RFC 5050*, Nov. 2007.

[16] Internet Research Task Force, "Delay-Tolerant Networking Research Group," 2013. [Online]. Available: http://www.dtnrg.org

[17] G.-S. Ahn, A. Campbell, A. Veres, and L.-H. Sun, "Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN)," *IEEE Trans. on Mobile Computing*, vol. 1, no. 3, pp. 192–207, Jul 2002.

[18] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE Journal on Selected Areas in Comm. (JSAC)*, vol. 19, no. 7, pp. 1300–1315, Jul. 2001.

[19] C.-C. Chen, G. Tahasildar, Y.-T. Yu, J.-S. Park, M. Gerla, and M. Sanadidi, "CodeMP: Network coded multipath to support TCP in disruptive MANETs," in *IEEE Int. Conf. on Mobile Adhoc and Sensor Systems (MASS)*, Las Vegas, USA, Oct. 2012, pp. 209–217.

[20] D. Lucani, M. Stojanovic, and M. Médard, "Random Linear Network Coding for Time Division Duplexing: When to Stop Talking and Start Listening," in *IEEE Conf. on Computer Comm. (INFOCOM)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1800 –1808.

[21] K. A. Harras and K. C. Almeroth, "Transport Layer Issues in Delay Tolerant Mobile Networks," in *IFIP Networking*, Coimbra, Portugal, May 2006, pp. 463–475.

[22] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, "Performance Modeling of Epidemic Routing," *Elsevier Computer Networks*, vol. 51, pp. 2867–2891, 2007.

[23] E. Bulut, Z. Wang, and B. Szymanski, "Cost-effective multiperiod spraying for routing in delay-tolerant networks," *IEEE/ACM Trans. on Netw.*, vol. 18, no. 5, pp. 1530–1543, 2010.

[24] R. Groenevelt and P. Nain, "Message delay in MANETs," in *ACM SIGMETRICS*, Banff, Canada, Jun. 2005, pp. 412–413.

[25] A. Hanbali, A. Kherani, and P. Nain, "Simple models for the performance evaluation of a class of two-hop relay protocols," in *IFIP Networking*, Atlanta, USA, May 2007.

[26] H. Cai and D. Y. Eun, "Crossing over the bounded domain: From exponential to power-law inter-meeting time in manet," in *ACM Conf. on Mobile Computing and Networking (MOBICOM)*, Montreal, Canada, Sept. 2007.

[27] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnovic, "Power law and exponential decay of inter contact times between mobile devices," in *ACM Conf. on Mobile Computing and Networking (MOBICOM)*, Montreal, Canada, Sept. 2007.

[28] U. Lee, S. Y. Oh, K.-W. Lee, and M. Gerla, "Scaling property of delay tolerant networks in correlated motion patterns," in *ACM SIGCOMM Workshop on Challenged Networks (CHANTS)*, Barcelona, Spain, Aug. 2009.

[29] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multi-copy case," *IEEE/ACM Trans. on Netw.*, vol. 16, pp. 77–90, Feb. 2008.

[30] E. Altman, T. Basar, and F. De Pellegrini, "Optimal monotone forwarding policies in delay tolerant mobile ad-hoc networks," *Elsevier Performance Evaluation*, vol. 67, no. 4, p. 299–317, 2010.

[31] T. G. Kurtz, "Solutions of Ordinary Differential Equations as Limits of Pure Jump Markov Processes," *Journal of Applied Probability*, vol. 7, no. 1, pp. 49–58, 1970.
[32] M. Benaïm and J.-Y. Le Boudec, "A class of mean field interaction models for computer and communication systems," *Elsevier Performance Evaluation*, vol. 65, no. 11-12, pp. 823–838, 2008.
[33] Y. Lin, B. Li, and B. Liang, "Efficient network coded data transmissions in disruption tolerant networks," in *IEEE Conf. on Computer Comm. (INFOCOM)*, Phoenix, USA, Apr. 2008, pp. 1508–1516.
[34] K. Price and R. Storn, "Differential Evolution: a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimiz.*, vol. 11, pp. 341–359, 1997.
[35] L. Sassatelli and M. Médard, "Inter-session network coding in delay-tolerant networks under spray-and-wait routing," in *IEEE Int. Symp. on Modeling and Optimization of Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Paderborn, Germany, May 2012, pp. 103–110.
[36] J. Burgess, B. Gallagher, D. Jensen, and B. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," in *IEEE INFOCOM*, vol. 10, Barcelona, Spain, Apr. 2006, pp. 1–11.

Lucile Sassatelli received the M.Sc. and the Ph.D. degree in telecommunication engineering from the University of Cergy-Pontoise, France, in 2005 and 2008, respectively. From November 2008 until September 2009, she was a Postdoctoral Fellow in the Reliable Communications and Network Coding Group (RLE) at Massachusetts Institute of Technology (MIT), Cambridge, USA. Since September 2009, she has been an Assistant Professor at the University Nice Sophia Antipolis, France. Her research activities focus on network coding, mobile and delay-tolerant ad hoc networks and social networks. Dr. Sassatelli has been in the technical program committee of IEEE RAWNET/WNC3 2009 and 2013, WCSP 2012 and VTC 2013.

Arshad Ali received a B.Sc. in Mathematics and Physics from Punjab University, Lahore in 1997. In 2003, he earned a M.Sc. degree in Computer Science from Punjab University, Lahore. Next in 2009, he received a Master diploma in Information Technology (Informatique) with speciality in Mobile Networks. He completed his Ph.D. degree in Information Technology, Telecommunications and Electronics jointly with Institute of Telecom SudParis and UPMC (Paris VI) in November 2012. Between 2005 and 2008, he served as an Audit Officer in the office of Auditor General of Pakistan. After PhD, he worked on energy efficiency for LTE networks in the radio engineering for mobile networks team of Orange labs till November, 2013. Since December 2013, he is working as Assistant Professor at University of Lahore, Lahore, Pakistan.

Manoj Panda obtained a PhD from the Indian Institute of Science Bangalore and MTech from the Indian Institute of Technology Kanpur. Manoj Panda is a researcher at the Swinburne University of Technology, Australia, from 2012. His research interests lie in Intelligent Transportation Systems and Performance Analysis of Computer Communication Systems.

Tijani Chahed holds BS and MS degrees in Electrical and Electronics Engineering from Bilkent University, Turkey, and PhD and Habilitation a Diriger des Recherches (HDR) degrees in Computer Science from the University of Versailles and the University of Paris 6, France, respectively. He is currently a Professor in the Telecommunication Networks and Services department in Telecom SudParis, France. His research interests are in the area of quality of service and teletraffic engineering, both in the wireline and wireless networks.

Eitan Altman received the B.Sc. degree in electrical engineering, the B.A. degree in physics, and the Ph.D. degree in electrical engineering from the TechnionIsrael Institute of Technology, Haifa, Israel, in 1984, 1984, and 1990,respectively, and the B.Mus. degree in music composition from Tel-Aviv University, Tel-Aviv, Israel, in 1990. Since 1990, he has been a Researcher with the National Research Institute in Informatics and Control (INRIA), Sophia-Antipolis, France. He is on the Editorial Boards of several scientific journals.Dr. Altman has been the (co)chairman of the program committee of international conferences.He received the Best Paper Award in the Networking 2006 conference and is a coauthor of two papers that have received the Best Student Paper awards at QoFis 2000 and Networking 2002. In October 2012, Dr. Altman has been awarded the prize France-Telecom of Académie des Sciences.