Inter-Session Network Coding-based Policies for Delay Tolerant Mobile Social Networks

Neetya Shrestha and Lucile Sassatelli

***Abstract*—We consider Delay Tolerant Mobile Social Networks (DTMSNs), which are opportunistic networks made of human-carried wireless devices clustered into social communities. In such environments, routing is a challenge as the limited resources (like memory and contact opportunities) must be efficiently used and shared between the sessions (or users, contents). To handle several unicast sessions, Inter-Session Network Coding (ISNC) has been proven necessary for optimal throughput in general networks, but is a delicate problem as it can quickly get detrimental. This paper investigates that ISNC can be beneficial in DTMSNs when used on top of a social-aware routing algorithm, whereas we exemplify and make explicit why any gain can hardly be expected with greedy replication, in regard to the current literature on ISNC. We then design decentralized criteria to control when and where in the network ISNC should be triggered, based on the nodes features (buffer size and social relationships) and network current load. These criteria are tested extensively on real-world contact traces, in terms of various metrics such as number of deliveries, mean delay or fairness. Our online ISNC protocol builds on the SimBet utility-routing policy. Our ISNC protocol can however run on top of any social-aware routing.**

I. Introduction

We consider intermittently connected networks made of human-carried wireless devices, and whose physical meeting patterns make cluster into social communities. We abbreviate them by Delay Tolerant Mobile Social Networks (DTMSNs). The three main goals of DTMSNs in civilian applications can be deemed as: (i) to provide network access to remote communities (e.g., Bytewalla [2]), (ii) provide cheaper content access by file exchange in ad hoc mode (e.g., PSN [3], [4], Liberouter [5]), (iii) to offload the cellular networks (e.g., [6], rescue operations).

In such disrupted, energy and memory-constrained environment, routing is a challenging task. In order not to flood the network with copies of the same packet, incurring maximum energy expenditure and memory load, it is provably better to leverage the social features of the underlying connection graph (so-called social-aware routing). We can cite BubbleRap [7] and SimBet [8], where some global and local ranks are used for each node to orientate and control the spreading. In [9], Mtibaa et al. present PeopleRank that defines the rank based on the PageRank Web algorithm.

To improve the probability of delivery within a certain deadline, several copies of the same packet can be disseminated, and this benefit may further improve with Network Coding (NC) which has attracted an increasing interest for DTNs [10]. NC is a networking paradigm that is a generalization of routing [11], aiming in particular at improving throughput and resilience to topology changes. There are two versions: (i) intra-session NC, mixing only the packets of the same session, (ii) inter-session NC (ISNC) mixing packets possibly pertaining to different sessions. ISNC is necessary to achieve optimal throughput in general (see [12] and references therein), but is a delicate problem because to retrieve its intended packets, a destination node needs to receive also other sessions' packets, called 'remedy packets' thereafter (as in [13]). If it does not, mixing sessions can degrade performance as compared to routing.

The object of this article is to show that ISNC can bring some gain in DTMSNs with unicast sessions, and designing online social-aware ISNC policies. For multiple unicast sessions in directed networks, the NC gain (in throughput) compared to plain routing has been proven unbounded [14], but the optimization problem has been shown NP-complete [11]. A number of works (e.g., [12], [13]) have come up with approximate solutions for static directed networks. When coming to DTNs, there is a priori no reason for considering that two nodes can exchange packets in a single direction. For multiple unicast in undirected networks, [15] conjectures that ISNC does not improve throughput over routing, which remains an unsolved question to date. While this has been proven for certain classes of networks (bipartite and planar graphs [15], [16]), upper-bounds on the gain have been obtained for some other classes (e.g., $9/8$ for fully reachable networks recently [17]). The non-directionality of DTNs is hence a first hurdle to the possible gain with ISNC. However, ISNC has proven very attractive in undirected wireless mesh networks [18], specifically owing to the time-shared wireless medium. So one can think that the constrained shared resources (buffer, contact opportunities) in DTNs can make ISNC attractive too, despite the non-directivity. But the second difficulty to readily apply this reasoning to DTNs is that there is no radio interference owing to the low node density and radio range.

Our contributions are threefold:

• We first provide a detailed discussion on the gains which can be expected from resorting to ISNC in DTMSNs, in regard to the current state of knowledge on ISNC in undirected networks. We make the hypothesis that the sub-optimality of social-aware routing, with respect to optimal routing for a certain known graph topology and traffic matrix, may benefit from ISNC to better handle competing unicast sessions. We provide an example showing this is indeed the case, thereby

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

Neetya Shrestha and Lucile Sassatelli are with Université Côte d'Azur, CNRS, I3S, Sophia Antipolis, France. Emails: {shrestha,sassatelli}@i3s.unice.fr.

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

supporting our approach. In particular, we discuss the interplay between Buffer Management (BM) and coding gains.

• We then design decentralized coding criteria that allow to trigger ISNC when it may be beneficial, based only on local information gathered by the mobile nodes. These criteria take into account the copy budget, network load and underlying social structure. Specifically, we build on the recent modeling results of [19], [20] and wield them to derive low-complexity delay estimations.

• These criteria are then used on top of the SimBet [8] social-aware routing policy, and are extensively tested on various real-word traces, compared against the performance of plain routing, for several metrics (among which rate, delay, fairness). ISNC triggered with our criteria is shown to bring gains in several cases, and we study how these gains vary over the copy budget and the network load. We emphasize that these ISNC policies can be used on top of any routing policy.

II. Related works

For homogeneous DTN, several works have considered intra-session NC which is now well understood in this case. Lin et al. in [10] investigated the use of intra-session NC using the Spray-and-Wait algorithm (SaW) [21] and analyzed the performance in terms of the bandwidth of contacts, the energy constraint and the buffer size. In [22], NC is considered at some intermediate hub nodes, but only across packets destined to the same destination node. Here, we tackle the more general problem of control of pairwise ISNC for unicast sessions with different destinations. In [23], Zhang *et al.* consider both intra- and inter-session NC in homogeneous DTNs. For unicast sessions with different sources and destinations, uncontrolled IS-NC is shown not to perform better than intra-session. In [24] we have presented an ISNC policy and its analytical model to express the optimization problem. The number of packets per session can be arbitrary, corresponding to the case where a file is split into several packets, and the metric (whether it be delay or delivery probability) is on the whole file. Also, 2 sessions were considered. To tackle the optimization problem of ISNC, we depart from this approach by choosing to reduce the parameter space (considering a single message/packet per session, as done in, e.g., [7], [8], [9]) to identify sound heuristics and design decentralized coding criteria. This article is also the only one where we study the problem of determining whether and how ISNC can actually be beneficial in DTMSNs, with both an analysis of existing theoretical results obtained so far, and detailedly analyzed experiments on small topologies.

III. Network model

To first reason on the possible gains of ISNC in DTMSNs, then to devise coding criteria aimed at wisely triggering ISNC, we consider the following network model. Though, it is worth noting that this model does not restrict the applicability of the so-devised ISNC algorithms which are tested on real-world traces in Sec. VI. We consider a network made of N nodes grouping into C communities. We assume that the number of meetings per unit of time between two given nodes is invariant over time and Poisson distributed, according to the findings of [25]. The average of this distribution is named inter-meeting intensity. We consider that all nodes pertaining to the same community i have the same inter-meeting intensity β_{ij} towards any other node of community j. The concept of community imposes that $\beta_{ii} > \beta_{ij}$, for all $i \neq j$. We consider the network bears R unicast sessions with source-destination node pairs (S_i, D_i), $i \in \{1, R\}$. A session is made of $K = 1$ packet (or message), and $P_i(\tau)$ denotes the probability that D_i has obtained its intended packet by time τ. Let $U(.)$ be any classical utility concave function, taken as $\log(1 + x)$ here. If $R = 2$, then we defined the utility over both sessions as $obj(\tau) = U(P_1(\tau)) + U(P_2(\tau))$. The nodes' buffer size is denoted by B (in packets); we take $B = 1$ in this article. We denote by "bandwidth" Bw the number of packets which can be sent in each direction upon each meeting. We take $Bw = 1$. General settings are discussed in Sec. V-E. We later on study the network load by making vary R while keeping the buffer size and bandwidth constant.

We build on the buffer structure for intra-session NC framework employed with SaW [10]. At a relay node buffer, a packet is associated with 3 fields: index, spray-counter, payload. When a packet is simply replicated upon node meeting, its index and payload are copied, and the spray-counters are updated at both nodes (binary sharing in SaW [21]). When the receiving node's buffer is full, if its packet $P^{(1)}$ payload is overwritten by the sum (XoR or in a higher order Galois field) with received packet $P^{(2)}$, the index is changed to denote the packet is coded (named $P^{(3)}$ for $R = 2$ in the next section), and the spray-counters are updated as described in Sec. IV-C. The initial (maximum) spray-counter is denoted by M and called the copy budget. Thereafter, "utility-based" or "social-aware" routing algorithms refer to policies which flow packets through relays with high utility towards the destination. We will employ the SimBet algorithm [8], but our ISNC framework can be used with any multi-copy algorithm. With SimBet, the copy budget shared upon replication is proportional to the meeting nodes' utilities [8, Sec. 4.4]. When the spray-counter at receiver is below 1, it is not transmitted, if it is at sender, it drops this packet once transmitted.

IV. ISNC gains and impacting factors

A. Impact of the routing algorithm

Determining whether ISNC can be beneficial to transfer multiple unicast sessions in undirected networks remains to this day an unsolved problem, which is known as the *multiple unicast network coding conjecture* [15]. This conjecture states that in an undirected network with multiple unicast sessions, network coding does not lead to any coding advantage over routing. This could be verified for some graph families (e.g., bipartite and planar [16], [15]). For some other families, the upper-bound of 9/8 on the coding gain (in throughput) has been recently proven [17]. It is worth noting that this problem for multiple unicast is closely related to that for multicast, for which the coding gain has an upper bound of 2 for general undirected networks (for half-integer routing, even less for

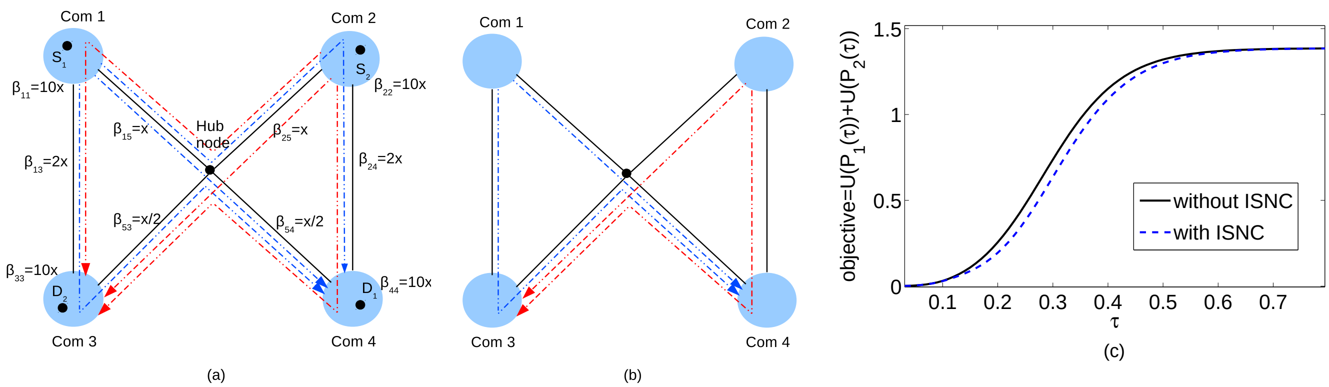

Fig. 1. (a) All possible routes are depicted in blue and red for each session 1 and 2, respectively. (b) The set of single routes allowing to get a gain with ISNC. (c) Epidemic routing on the butterfly network.

fractional routing), tightened to 9/8 for combination network coding [26]. The authors of [26] have also interestingly shown that the reduction in routing cost brought by coding is bounded with the same upper-bound as the gain in throughput, thereby suggesting ISNC may not help in saving link usage if it cannot improve throughput.

As well, a DTMSN can be regarded as a weighted graph where a mobile node is a graph node, and the weight on each edge is a combination of β_{ij}, B and Bw [27]. Considering the meeting duration can be arbitrarily split for transmissions in both directions, a DTMSN is hence a general undirected network where deciding upon fractional routing means deciding on buffer management and scheduling. Given the works mentioned in the above paragraph, solving for best (for a known graph topology) routing, coding, buffer management and scheduling in DTMSNs can be conjectured not to outperform the counterpart solution without ISNC. We exemplify this conjecture in Example 1 below.

However, the coding gains mentioned above refer to the optimal code and route assignments and their outcome in terms of throughput and link cost. Owing to the inherent uncertainty of DTMSN environments, the routes and schedules must be chosen locally at the nodes with a mechanism called *utility-based routing* (proven in [28, Sec. 3.2]). The utility is either assigned to nodes [29], [8] or messages [28]. In order to make up for the so-obtained suboptimal routing policies bearing several unicast sessions, allowing ISNC atop utility-based routing is an alternative worth investigating, as the sub-optimality (with respect to unknown traffic matrix and topology) of the utility-based routing can turn ISNC into beneficial, as exemplified below. Because of its ability to leverage various levels of social-network analysis, we choose SimBet [8] as the underlying utility-based routing, though our approach can be used with other utility-based policies.

Example 1: We consider the well-known butterfly topology represented in Fig. 1, made of $C = 5$ communities, with $N_i = 250$ nodes for $i \in \{1, 4\}$ and the fifth community is a single hub mobile node. We consider two unicast sessions whose source-destination node pairs are in communities $(1, 4)$ and $(2, 3)$, respectively. For all $i, j \in \{1, C\}$, β_{ij} are those indicated in Fig. 1 where $x = 5.10^{-3}$, and they are chosen such that the remedy packets $P^{(1)}$ and $P^{(2)}$ get faster than $P^{(3)}$ to D_2 and D_1, respectively. So being the expected point of congestion, we set the hub node to mix both sessions' packets as soon as it can. In order not to impede its delivery, we make $P^{(i)}$ overwrite any other packet in C_{D_i}, $i = 1, 2$.

We run the first experiment with a greedy replication algorithm, Spray-and-Wait [21], that spreads the copy budget (set to $M = 2000$, i.e., epidemic) by handing redundancy (packet replicas or coded packet) to the first met nodes (regardless of any social feature). Despite the underlying butterfly topology of the social structure and the limited buffer size (of 1 packet), the outcome (represented in Fig. 1.c from our analytical model of [24] adapted to this simple case) shows no improvement in objective function $obj(\tau)$ with ISNC with respect to no ISNC. The reason is that the routes taken by the packets can be verified to be those shown in Fig. 1.a, that are routes exploiting the bi-directionality. This observation is in line with the aforementioned conjecture which Langberg and Médard formulated informally as "Undirecting the edges of [a graph] is as strong as allowing network coding" [30]. Indeed, in DTMSNs we deal with two levels of undirectionality: (i) first, a contact between two mobile nodes is (half-)duplex; (ii) second, at the community level, a non-social aware routing algorithm permits the packets of a given session to flow in both directions between both communities. ISNC at the hub node might bring some gain if the routes taken by the sessions were those depicted in Fig. 1.b, which happens to be the case: Fig. 2.a depicts the evolution of the number of each packet type in each community under SimBet routing [8]. Packet $P^{(1)}$ first spreads inside its source community $C_{S_1} = 1$, then reaches $C_{D_1} = 4$ mostly through the hub node as we see that the increase in $P^{(1)}$-infected nodes in $c = 4$ precedes the increase in $c = 2$, while the hub node gets readily infected. Community $c = 2$ remains almost uninfected by $P^{(1)}$. This shows that the routes taken by $P^{(1)}$ and governed by SimBet are very close to those in Fig. 1.b, identified as the routes susceptible to benefit from ISNC. This supports our motivation that ISNC can improve performance of multiple unicast sessions routed with a social-aware routing algorithm.

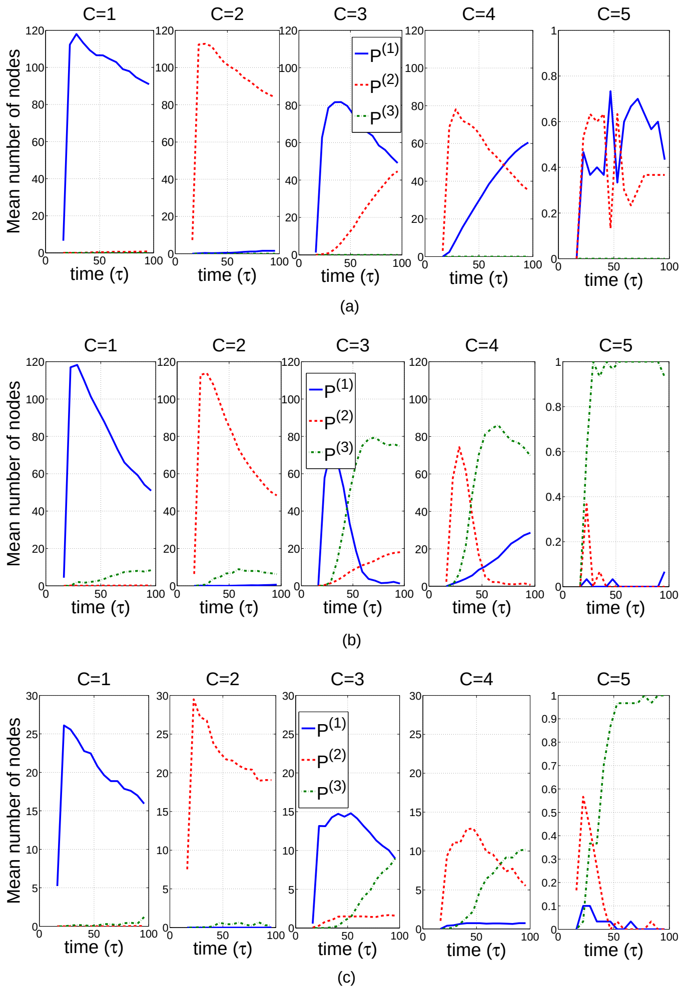

Fig. 2. Mean number of nodes infected with each type of packets in each community, with SimBet routing. (a) without ISNC, $M = 200$, (b) with ISNC, $M = 200$ and (c) with ISNC, $M = 40$

B. Combining social-aware routing and ISNC

We now present the questions to answer to enable ISNC on top of a social-aware routing algorithm. To do so, two distinct, yet correlated, problems arise:

P1 Where and when should ISNC be triggered in the network?

P2 How to have the destination nodes, whose packets have been inter-session coded, receive the remedy packets?

Problem P1 corresponds to the choice of which mobile node, when presented with a coding opportunity, should mix two packets of different sessions. In the remainder of this section, we consider the butterfly topology and set this coding node fixed to the hub node, which will mix packets as soon as it can. This is done in order to assess what are the impacts of main network components on the ISNC gain (BM, copy budget and network load), to then be able to devise ISNC criterion to wisely trigger ISNC only when a gain can be expected.

Problem P2 is tied to P1 as session mixing must be triggered only if the destination nodes receive fast enough the remedy packets which will allow them to retrieve their intended packet earlier than with no ISNC. There are two main ways to deal with P2. First, one can think about some signaling mechanism that informs the source node S_1 to send its packet to destination D_2 (and symmetrically for S_2 and D_1), which is not the normal mode of operation since D_2 is not the intended destination of S_1. Examples (not for DTMSNs) include [12] or [13] where the so-called antidote request is issued with this aim. The second solution, as in the now well-known application of ISNC for ad hoc wireless mesh networks [18], is to leverage the other sessions' packets which have been overheard opportunistically, without being explicitly sent to other destinations. The equivalent of such strategy in DTMSNs is to take benefit from packets reaching other sessions' destinations, owing to the routing choices of the social-aware routing algorithm. Despite the fact that such strategy may make miss some coding opportunities, this is the strategy we choose in this work to avoid additional signaling costly in DTMSNs.

The next three sections aim at identifying how key network parameters impact the ISNC performance. This is the necessary preliminary step to design ISNC criteria in Sec. V, that must take these parameters into account. In particular, as ISNC is meant to handle the network load, hence the sessions' competition to access resources, a relevant choice of BM is crucial in comparing fairly ISNC with routing. This is presented and discussed in Sec. IV-C and IV-F. The impact of copy counter and number of sessions is investigated in Sec. IV-D and IV-E, respectively.

C. Buffer management and copy counters

For this proof of concept, we keep set in this section that the only node allowed to mix packets is the hub node, and it does so as soon as it can, i.e., a XoRed packet named $P^{(3)}$ can be generated at node H only if: (i) H already holds $P^{(1)}$, and has no more room in its buffer, and (ii) H meets A holding $P^{(2)}$ (or the other way around), and (iii) SimBet triggers the transmission of $P^{(2)}$ to H, based on utilities only, and (iv) the BM below allows the replacement of $P^{(1)}$ by $P^{(3)}$ at H. The copy counter assigned to $P^{(3)}$ is then the sum of the counter of the replaced packet and the copy budget handed over by A, determined by the copy share of SimBet. The details are provided in Appendix C. We consider a BM which cannot favor ISNC compared with single routing and bias the results (the symmetric holds for session 2):

F1 Destination node D_1 can erase $P^{(1)}$ from all nodes (in any community) but cannot erase $P^{(2)}$. At a node n in community C_n, a coded packet can replace an uncoded packet if C_n is neither the source nor destination of the uncoded packet, but the destination community of one of the mixed packet, and an uncoded packet can replace a coded packet as well under symmetric condition.

F2 Keep on spreading the energy budget even though the payload of the already-there packet does not change. For example, when node A with $P^{(1)}$ meets B with $P^{(3)}$, if SimBet utility would trigger transmission of some copy share to B, then it gets added to the counter at B, although the payload remains the same. This feature, allowed by the use of ISNC, allows to re-focus the copy budget through coded packets to better serve both sessions.

F3 Destination node D_1: (i) erases $P^{(3)}$ upon reception from a node in C_{D_1}, and (ii) signals to the nodes of outside communities that it has received $P^{(3)}$ and/or recovered $P^{(1)}$.

We consider the topology and sessions of Fig. 1.a., with $\beta_{13} = \beta_{24} = x$, $\beta_{15} = \beta_{25} = 10x$, $\beta_{53} = \beta_{54} = 8x$ and $\beta_{cc} = 15x$, for $c = 1, \dots, 4$, with $x = 5.10^{-4}$. We use 15% of the simulation duration as warm-up phase to provide an opportunity to gather information about the nodes within the network, as in [8]. After the warm-up phase, the messages are allowed to disseminate in the network. We plot in Fig. 3 the objective function $obj(\tau)$ (Sec. III). In order to assess the impact of each component F1, F2 and F3, the results incrementally adding each of those are shown. The curve label "without IS" refers to the case where SimBet routing alone is used, without any session mixing. In this case, only F1 can apply. We point out that here intra-session NC is not mentioned as messages are made of a single packet. We show that with such a systematic coding at the hub node in the butterfly topology operated with SimBet, ISNC outperforms single routing without specific additional buffer or copy counter management. The gain is improved over F2 and F3. The ISNC gain is in particular explained by Fig. 2.b, where we can see that the hub gets occupied by a coded packet ($P^{(3)}$) systematically. This hub node is indeed the point of congestion, as it is at crossroads and has buffer size of 1.

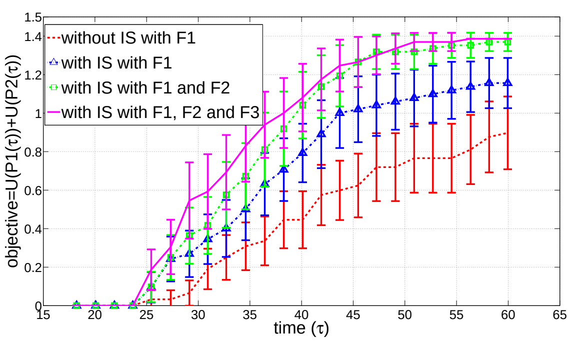

Fig. 3. Objective value for the different buffer and copy counter management (2 crossed sessions).

D. Impact of the copy budget on ISNC gains

We now make vary the copy budget M and look at the impact on the coding gain, both in terms of number of delivered messages and mean delay. Without ISNC, the message delivery delay is the time for the intended destination to get the first copy. With ISNC, the message delay is the time for the destination node to recover its intended message, either by receiving the original (uncoded) packet, or by receiving a coded packet and its remedy packet (then the delay is the maximum of both reception delays). There are still only 2 unicast sessions as depicted in Fig. 1. Fig. 4 shows results for $M = 200$ and $M = 40$. We observe that the ISNC gain decreases with M. We hypothesize that this stems from the lower spreading of remedy packets when M decreases. We verify it with Fig. 2.c, where we can see that in these communities, the maximum number of $P^{(1)}$ (or $P^{(2)}$ in com. 4) goes roughly from 120 for $M = 200$ to 30 for $M = 40$.

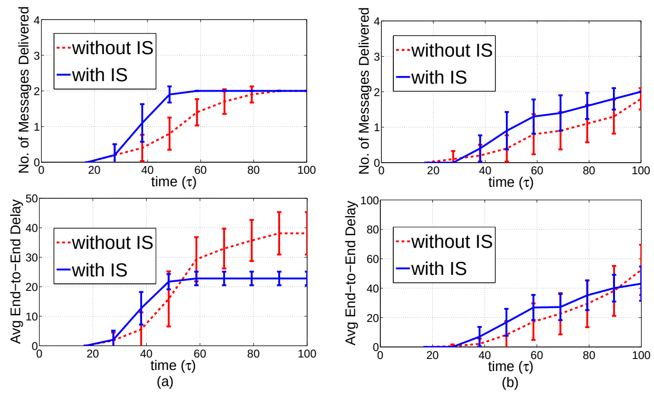

Fig. 4. $R = 2$, crossed sessions, (a) $M = 200$, (b) $M = 40$

E. Impact of the network load on ISNC gains

We now look at the impact on the ISNC gain of the network load, taken as the number of concurrent sessions. We emphasize that we consider only pairwise ISNC, that is any coded packet cannot stem from the linear combination of more that 2 different sessions. In order to avoid the network to get blocked up owing to too many packets occupying the nodes' buffers and not being dropped fast enough, we add an extra feature to the BM presented in Sec. IV-C:

F4 (i) if the buffer is full, the packet can be overwritten by the incoming one with a certain probability that depends on the counters of each packet (if packet B competes with packet A already in the buffer, B overwrites A with probability $\frac{0.8 * n_B}{n_A + n_B}$, where n_A and n_B are the copy counters of A and B), and (ii) each time a packet (either replica or coded) is created at a node, it is assigned an exponentially distributed TTL whose mean decreases with its copy counter (this mean is taken as $V_0 \frac{current\_counter}{M}$, with V_0 taken as the trace duration in the experiments).

Fig. 5 show results for 4 sessions, while Fig. 6 show results for 10 sessions. For Fig. 5 and 6.a, half of the source-destinations pairs are picked out in communities $(1, 4)$, the other half in communities $(2, 3)$. We observe that, for a given M the ISNC gain decreases with the number of sessions. There are a lot of coding opportunities at the hub node as a lot of packets compete to access its buffer, but because of the higher load (with no more bandwidth or buffer resources), the required remedy packets do not arrive fast enough at the destinations nodes to allow them to decode the coded packet faster than with mere routing. Fig. 6.b shows results when the source-destinations pairs are picked uniformly at random in the four side communities. We observe in this case that the degradation entailed by systematic ISNC at the hub node reduces compared with the case of all crossed sessions. This is explained by the fact that there are less sessions that are crossed and hence get their packet coded at the hub, that thereby increases the packet recovery at these destinations. The network load hence impacts the ISNC gains through the average relay occupancy it yields, and which depends on the concurrent sessions, the buffer size and the copy budget.

We thereby identified parameters which must be taken into account when choosing whether to generate inter-session

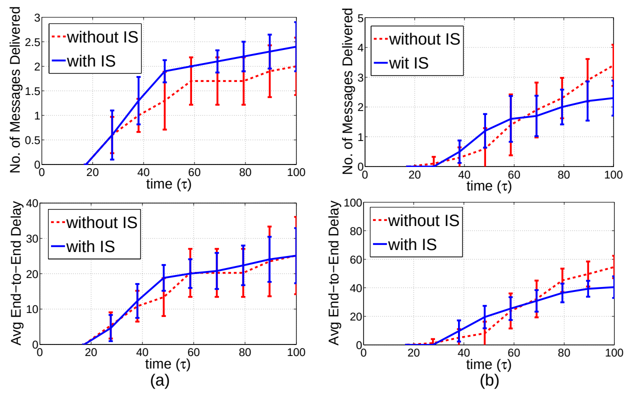

Fig. 5. $R = 4$, crossed sessions, (a) $M = 200$, (b) $M = 40$

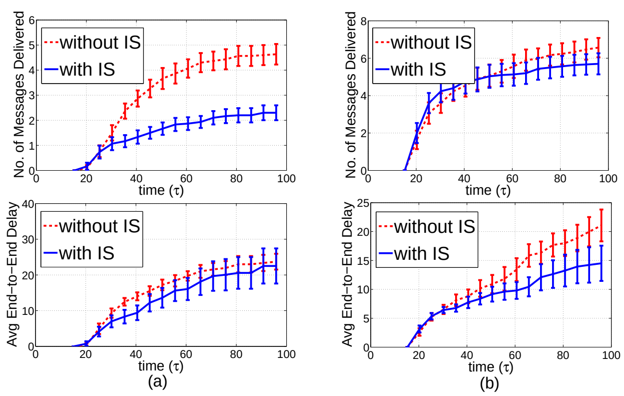

Fig. 6. $R = 10$, $M = 200$, (a) crossed sessions, (b) homogeneous sessions.

coded packets. In particular, it is crucial to endeavor to make ISNC never perform worse that plain routing by wisely triggering session mixing upon criteria encompassing these conditions. This is the object of Sec. V.

F. Discussion on buffer management

Let us specify that the above F1−F4 items constitute BM choices aimed at fairly comparing ISNC with non-ISNC policies. We can also consider the scheme proposed by Krifa et al. in [31] whose tractability, for DTMSNs, is based on the assumption that the node inter-meeting intensities stem from the same distribution, and the copies are spread over numerous enough different nodes. We could envision such a policy to replace F1, and hence apply both to no ISNC and ISNC policies. However doing so would require to overcome a number of hurdles: (i) that BM does not depend on the IDs of packet's source and destination, as the drop/scheduling probability is the same at all relay nodes. So if this BM were to be implemented, it would make sense to keep the per-community BM presented above. But only making the BM designed in [31] depend on the communities is challenging. (ii) To implement this BM of [31] with ISNC, the utility functions maximized in [31] would need to be redefined to include coded packets, which is not straightforward. That is why for the sake of interpretation, we did not implement this policy, even only for uncoded packets, in order not to introduce any uncontrolled unfairness with a BM policy whose assumptions do not hold for the highly heterogeneous topologies and constrained copy budget considered here.

V. Design of the decentralized coding criteria

In this section, we address the problem of deciding whether to trigger ISNC, and if so, where and when in the network, that is mixing sessions based only on local information gathered at the nodes. We emphasize that we consider only pairwise ISNC, so a coded packet can mix at most two sessions.

A. Principle and approximation framework

We consider the decision problem of mixing session $Z_1 = (S_1, D_1)$ with session $Z_2 = (S_2, D_2)$ at node n_c. $P^{(1)}$, $P^{(2)}$ and $P^{(3)}$ still denote the session packets and the XoR of both, respectively. It can be ensured that ISNC triggered at node n_c does not perform worse than no ISNC if and only if:

$$\begin{cases} Delay(S_2 \to D_1) < Delay(S_1, S_2 \to n_c) + Delay(n_c \to D_1) \\ \text{AND} \\ Delay(S_1 \to D_2) < Delay(S_1, S_2 \to n_c) + Delay(n_c \to D_2). \end{cases} \tag{1}$$

where $Delay(S_2 \to D_1)$ is the delay for D_1 to get $P^{(2)}$, $Delay(S_1, S_2 \to n_c)$ is the delay for node n_c to get both $P^{(1)}$ and $P^{(2)}$, and $Delay(n_c \to D_1)$ is the delay for $P^{(3)}$ to make it to D_1, and symmetrically for the other inequality. This condition corresponds to code only if each remedy packet could make it to both destinations by the time the coded packet arrives. Another necessary condition for ISNC not to perform worse than plain routing is to ensure that the coded packet reaches the destination as fast as an uncoded packet, and if not, it is so in order one destination receives its intended source packet before the coded packet. This depends on the three following elements:

• Copy counter of the coded packet, if there is no congestion (same behavior as if storage and bandwidth were unlimited): the split of copy budget $p.counter(SimBet)$ (see Appendix C) ensures that at any node at any time, the counter associated with the coded packet is always at least equal to what the counter of the uncoded packet would be. Indeed, let L_1 and L_2 denote the copy counters of $P^{(1)}$ and $P^{(2)}$ at the time they get coded and FU_1 and FU_2 the fraction of utilities corresponding to each session for nodes A and B ($FU_1 = \frac{U_B(D_1)}{U_A(D_1)+U_B(D_1)}$ and $FU_2 = \frac{U_B(D_2)}{U_A(D_2)+U_B(D_2)}$, respectively). Upon each transmission, the budget handed over to B writes as $(L_1 + L_2) * (FU_1 + FU_2) > L_1 FU_1 + L_2 FU_2$. As this inequality holds over the successive transmissions, this justifies the above claim.

• Copy counter of the coded packet, if there is congestion: the counter higher for $P^{(3)}$ than what it would be for $P^{(1)}$ or $P^{(2)}$ ensures that $P^{(3)}$ gets a higher forwarding precedence.

• Overwriting probability: the last element determining the delay of the coded packet to get to the destinations is whether it is more likely to get overwritten. This is indeed the case, but as described in F1 (Sec. IV-C), only an uncoded packet involved in a coded packet can overwrite it in the destination community, thereby ensuring the recovery of the source packet is not delayed at the destination.

So the delay of the coded packet can be higher than that of the uncoded only so that one destination can recover the source packet earlier, thereby ensuring the performance of ISNC to be at least as good as that of plain routing.

In order to estimate each of the three quantities involved in the above inequalities, we make the following choices.

- $Delay(S_1, S_2 \rightarrow n_c) = \max\big(Delay(S_1 \rightarrow n_c), Delay(S_2 \rightarrow n_c)\big)$, where the two arguments of the maximum are estimated by time counters kept in the packets $P^{(1)}$ and $P^{(2)}$'s headers.
- The estimation of $Delay(S_2 \rightarrow D_1)$ is done in different ways leading to the different coding criteria of the next section.
- As it can been seen in the next sections, the estimates we have on delays are only upper-bounds (owing to Eq. (2) and the subset of routes considered in the various criteria). That is why these upper-bounds are used as estimates of $Delay(S_2 \rightarrow D_1)$ because it appears in the left-hand side of inequalities Eq. (1). The so-obtained inequalities involving the upper-bound are hence a sufficient condition for the original inequalities Eq. (1) to hold. However, ensuring the inequalities where $D(n_c \rightarrow D_i)$ would be replaced with its corresponding upper-bound would not provide a sufficient condition as $D(n_c \rightarrow D_i)$ appears in the right-hand side. Additionally, when expressing the upper-bound provided in Eq. (2), we consider only a subset of possible routes (as detailed in the different criteria), thereby loosening even more the inequality Eq. (2). This increases the risk that inequality in Eq. (1) be not satisfied if $D(n_c \rightarrow D_i)$ were estimated this way. That is why, by lack of a non-trivial lower-bound and at the risk of being too conservative in triggering ISNC, we choose to consider $D(n_c \rightarrow D_i) = 0$, then verified in the numerical simulations on real-world traces that such designed ISNC criteria already bring gain.

Having the right-hand side terms of condition Eq. (1) given by time counters stored in the packet headers, the key problem is the estimation of the left-hand side term $Delay(S_1 \rightarrow D_2)$ (and symmetrically $Delay(S_1 \rightarrow D_2)$). To estimate this delay, we build jointly on [19] and [20], both authored by Picu *et al.*, and come up with three different criteria.

In [19], the authors analyze the delay for a message to reach all the nodes when epidemic routing is employed on a fully heterogeneous DTMSN, that is where the inter-meeting intensities λ_{ij} are different for all node pairs (i, j). To do so, the network state evolution (what nodes are infected at each time instant) is modeled as a discrete time Markov chain (with state space Ω) whose transition probabilities depend on the meeting probabilities p^c_{ij} of each node pairs (i, j) at each time slot. Based on the analysis of this Markov chain, the authors prove that, under epidemic routing, the expected delay $E[D_{ep}]$ for all the N network nodes to get infected by a message initially spread out by a single source uniformly chosen over all the nodes is upper-bounded by:

$$E[D_{ep}] < \frac{2\log(N-1)}{N\Phi} \ , \tag{2}$$

where Φ denotes the conductance of the underlying contact graph (whose edges between vertices have weight p^c_{ij}) defined as

$$\Phi = \min_{S \in \Omega} \phi(S) \tag{3}$$

where S is a cut, the conductance of a cut is $\phi(S) = \frac{\partial(S)}{|S||\bar{S}|}$ and $\partial(S) = \sum_{i \in S, j \notin S} p^c_{ij}$ is called the edge boundary of the vertex set S. Let us provide a physical interpretation of this result, that we will use later on. The cut S realizing the minimum in the conductance formula, can be seen as the set of nodes that, when infected while the others are not, yield the lowest probability to infect any other node. It is therefore the state in which the dissemination process is stuck the longest time before being able to escape. Then the inverse of conductance represents the time to escape this state. This analysis presented in [19] hence considers unlimited replication (flooding), and no constraint on the buffer occupancy. We shall address these limitations in the criterion design.

In [20], the authors present their DTN-Meteo framework which also considers a fully heterogeneous DTMSN, that is not operated with flooding anymore but with a limited-replication utility-based algorithm choosing the relays so as to optimize a certain objective function (that depends on the task at hand, e.g., routing or content placement). The network state evolution is modeled with the same kind of Markov chain as above, except that the transition probability between state x and y is given by $p_{xy} = p^c_{ij} A_{xy}$, where A_{xy} is an indicator function which is 1 if the utility of state y is greater than that of state x. The expected completion delay of the task is then derived, for example the expected delivery delay of the message to one node or a set of nodes, from its initial spreading by a uniformly chosen source node [20, Theorem 2]. A weakness of DTN-Meteo with the ISNC framework we consider is how multiple-copy routing is taken into account, and our ISNC strategy is relevant only if several copies of the message are spread (so that a non intended destination has a chance to grab one). Indeed, it is assumed in DTN-Meteo that $M << N$, allowing to neglect the spreading time of the M copies from a single source in [20]. This assumption cannot hold when the network load increases, inducing the buffers to fill up and hindering the spreading, that is a regime important for ISNC which aims at better managing the network load.

To sum up, the above works [19], [20] have the following limitations to solve readily the problem of delay estimation: (i) the session under scrutiny is assumed to be the only one running in the network; (ii) the copy counter sharing is not considered managed by the node utility; (iii) the theoretical results obtained in [20], [19] are averaged over all possible source nodes.

We employ several workarounds which lead us to devise three different criteria taking into account the above constraints. To provide the mobile nodes with a low-complexity delay estimation, we consider the results of [19], whose upper-bound on the epidemic delay given in Eq. (2) above is attractive owing to its simplicity, and we arrange heuristically for being able to use it in our context. We make use of [20] to insert the utilities in the last two criteria. The direct application of Eq. (2) would require the computation of the conductance of the whole graph, therefore entailing a high complexity at the nodes. That is the reason why our strategy is to restrict the paths we consider the remedy packet can take from S_1 to D_2 to single or two-hop paths, where the hops are in terms of communities (not nodes). We hence consider the subgraphs entailed by these restricted paths to compute the propagation delay of the remedy packet, using Eq. (2) with the number of nodes and conductance appropriately redefined on these subgraphs.

It is worth noting that considering a restricted number of routes provides an upper-bound on the upper-bound in Eq. (2). Using a looser upper-bound for the remedy delay makes the criterion conservative, rather missing coding opportunities than triggering ISNC when it should not.

B. Criterion 1

The first criterion we devise stems from considering the remedy can only go through one community-hop between source and destination. The delay estimate writes as

$$Delay_1(S_1 \to D_2) = \frac{2\log\left(N_{eff}-1\right)}{N_{eff}\Phi} \; .$$

We denote by N_{eff} the number of effective nodes involved in the subgraph we consider the remedy packet can travel. So, neglecting the limitations yielded by the load, N_{eff} would be $N_{C_{S_1}} + N_{C_{D_2}}$, since within one-community hop the remedy can be only either in community C_{S_1} or C_{D_2}. We take the network load (that makes certain nodes' buffers inaccessible to the session of interest) into account as follows. If all N nodes have a buffer size of B packets, and there are R sessions each with copy budget M, then $\min(1,(BN)/(MR))$ is the average buffer space available in the worst case, that is once all the sessions have spread entirely. This is a rough approximation as the distribution of the buffer occupancy may vary over the communities, so may the buffer size over nodes and copy budget over sessions. The general case is discussed in Sec. V-E. Then we take into account the limited replication by bounding the maximum number of nodes reachable by the packet: in the initial copy budget M (at source node S_1), only a fraction $f(U,S_1,D_2,D_1) = \frac{U(D_2\to D_1)}{U(D_2\to D_1)+U(S_1\to D_1)}$ (where $U(S \to D$ is the SimBet utility of node S towards node D) is estimated to be able to make it to the community of D_2. In the end, we come up with the following expression for N_{eff}:

$$N_{eff} = \min\left(N_{C_{S_1}}+N_{C_{D_2}}, f(U,S_1,D_2,D_1)M\min\left(1,\frac{BN}{MR}\right)\right) .$$

Having hereby included the utilities in the number of nodes that can relay the packet “epidemically”, we consider the edges of the subgraph have weight β_{ij}.

Lemma 5.1: The conductance Φ of a 2-community graph defined by β_{11}, β_{22} and β_{12} is expressed by:

$$\Phi = \min(\beta_{11},\beta_{22}) + \beta_{12} \; .$$

Proof: See Appendix A.

C. Criterion 2

The second criterion we design is a variation of the first one, where instead of including the utilities in the maximum number of accessible nodes, owing to the copy counter management based on SimBet, we incorporate the utilities as it is done in [20]: the node utilities come into the probabilities of handing the packet over to another node, and hence rather modulate the propagation speed rather than limit the number of accessible nodes based on the utilities. Only a one-community hop is still considered, and we have now for the parameters:

$$N_{eff} = \min\left(N_{C_{S_1}} + N_{C_{D_2}}, M\min\left(1,\frac{BN}{MR}\right)\right)$$

$$\Phi = \min(\gamma_{11},\gamma_{22})+\gamma_{12} \; , \text{ with } \gamma_{ij} = \beta_{ij}\frac{U(j\to D_1)}{U(j\to D_1)+U(i\to D_1)}$$

D. Criterion 3

The third criterion does not consider only a one-community hop for the remedy anymore, but instead considers the one-hop as well as all possible two-hop paths:

$$Delay_1(S_1 \to D_2) = \min(\Delta_{1hop}, \Delta_{2hops}) \; .$$

Δ_{1hop} is $Delay_1(S_1 \to D_2)$ obtained with Criterion 2.

$$\Delta_{2hops} = \min_{c\in\{1,C\}} \Delta^{(c)}$$

where c denotes all the possible intermediate communities between C_{S_1} and C_{D_2}. We express Δ_{2hop} similarly with

$$\Delta^{(c)} = \frac{2\log\left(N^{(c)}_{eff}-1\right)}{N^{(c)}_{eff}\Phi^{(c)}} \; .$$

For each Δ_{1hop} and Δ_{2hop}, we consider the respective associated subgraphs. That of $\Delta^{(c)}$ is made of communities C_{S_1}, C_{D_2} and c. With 3 communities, taking into account the impact of the copy counter sharing in the number of accessible nodes becomes tricky, and we resort to the approach of Criterion 2 and keep including the utilities in the subgraphs edge labels, rather than in the node number. Specifically, we still consider

$$N^{(c)}_{eff} = \min\left(N_{C_{S_1}} + N_{C_{D_2}} + N_c, M\min\left(1,\frac{BN}{MR}\right)\right) .$$

Lemma 5.2: The conductance Φ of a graph made of 3 communities C_{S_1}, C_{D_2} and c writes as:

$$\begin{aligned}\Phi^{(c)} \;\; = \;\; & \min\Big(\left(\gamma_{C_{S_1}C_{S_1}} + \gamma_{C_{S_1}C_{D_2}} - \gamma_{cc} - \gamma_{cC_{D_2}}\right)^- , \\ & \left(\gamma_{C_{D_2}C_{D_2}} + \gamma_{C_{S_1}C_{D_2}} - \gamma_{cc} - \gamma_{C_{S_1}c}\right)^- \Big) \\ & + \gamma_{cC_{S_1}} + \gamma_{cC_{D_2}} + \gamma_{cc} \; ,\end{aligned}$$

with $x^- = \min(x,0)$ and γ_{ij} defined in Criterion 2, for all $i,j \in \{1,C\}$.
Proof: See Appendix B.

E. Practical issues

a) Multiple sessions: Let $Z_n = (S_n, D_n)$ denote the source-destination pair of session n. We set, based on [10], that the sending node A schedules the packets to send out in the decreasing order of their respective copy counters. Then at the receiving node B, if no room is left, the received packet is checked for generating a coded packet by mixing it with an already present uncoded, which is chosen such that it maximizes the sum of differences of inequality terms in Condition 1. The ISNC policy is detailed in the general case in Appendix C.

b) Online parameter estimation: The network nodes run the online Community Detection (CD) algorithm of [32] (Modularity version) and SimBet utility computations in parallel. Then to implement ISNC as above, each node maintains matrices β and $\mathbf{U}$ (with β_{ij} and $U(i \to j)$ as component, i, j node indexes, size $N \times N$), and vector $C^{(vec)}$ (with $C_i^{(vec)}$ the community of node i, size N). These matrices are exchanged upon meeting of node A and B, and each node (B there) performs:

- Update $\beta(B)$ (resp. $\mathbf{U}(B)$) integrating $\beta(A)$ (resp. $\mathbf{U}(A)$) with an exponential weighted moving average.
- Increment $\beta_{AB}(B)$ (number of meetings per time unit depending on the trace).
- Once the update of the variables for the CD are made [32, Sec. 4.2], if A gets inserted into $C_B^{(vec)}$, then replace, $\forall j \leq N, \forall i \in C_B^{(vec)}$, β_{ij} with the average of the values.
- If $C_c^{(vec)}(A) \neq C_c^{(vec)}(B)$ for some node c, then set $C_c^{(vec)}(B)$ to the community with the highest number of nodes (owing to the merging process of CD).

The additional signaling and memory overhead of ISNC is therefore due to β, $\mathbf{U}$ and $C^{(vec)}$ and amounts to $2N^2 + N$ at most. This is what has been used in the simulations below. There are however a number of ways to reduce it. First, the order of overhead amount of the CD is n^2 ($F_{(approx.)}(j)$ [32, Sec. 4.2]), where n is the number of nodes met by a node. This can hence be close to N^2 depending on the topology, in which case the scaling of ISNC overhead in number of nodes is not different than that of CD. Second, like CD, the process may stop or update infrequently once it has converged, to save resource. Indeed, the involved quantity are stable in time (do not depend on the traffic matrix). Third, given that the above ISNC criteria consider routes restricted to a neighborhood of degree 2 as done in [8], [32], it can be investigated to keep information $\beta(B)$, $\mathbf{U}(B)$ and $C^{(vec)}(B)$ only for nodes c directly met by B, then the order would be back to n^2. Lastly, $\mathbf{U}$ may be deducible from β, possibly saving transfer and memory. As seen below, despite the online estimation of these variables, the numerical simulations (with the first 15%-period let as a warm-up phase described in Sec. IV-C) show that ISNC implemented this way bring some gain. Specifically, if we notice that the nodes liable to generate a coded packet are "hubs", i.e., relays at the crossing of different paths through at most 3 communities, it is indeed likely that these nodes get the information they need to trigger ISNC as fast as they get the legacy SimBet information to perform only routing.

The other parameters B, R and M are considered known at the nodes in the numerical simulations below, but they can be obtained with a moving average through the nodes, or in a decentralized manner like in [31, Sec. 4] (learning over number of messages, copies and possible TTL). B and M can also be default parameter in the application running ISNC.

VI. NUMERICAL ASSESSMENTS

We analyze the impact of ISNC on several metrics. The number of delivered messages, delay, number of hops and forwards are called "raw metrics" thereafter. A message is delivered successfully if the intended destination receives either a copy of the original packet, or a coded packet along with the right remedy. If a coded packet allows the destination to first retrieve the original packet, then the message delay is the maximum of respective delays to obtain the coded and the remedy. The same holds for the average Number of Hops (NH), counted in number of nodes the packet index has traveled through. When generating a coded packet, it is set to the maximum of the NHs of mixed packets. The total number of forwards is increased every time a message copy gets forwarded. For each of the above metrics $\mu(\tau)$ where τ is the time variable and T the simulation time span, we define the coding gain as $Gain_\mu = \sum_{\tau \leq T} \left(\frac{\mu_{w/IS}(\tau) - \mu_{w/oIS}(\tau)}{\mu_{w/oIS}(\tau)} \right)$. To assess the node fairness, we define the $Relativeload(n)$ of node n as $Relativeload(n) = load(n) - \frac{1}{N} \sum_{j=1}^{N} load(j)$ where $load(n)$ is the total number of forwards performed by node n over the whole simulation.

We run tests on four datasets: *Intel*, *Cambridge*, *Infocom05* and *MIT* collected from CRAWDAD [33]. These traces have been explored in several works such as [34], [8], [9], [7]. A summary of these datasets is provided in Table I. We refer the reader to [34] for further details.

Experimental datasets	Intel	Cambridge	Infocom05	MIT
Total devices	128	223	264	83
Network type	BT	BT	BT	BT
Scanning interval (sec.)	120	120	120	300
Duration (days)	3	5	3	30
No. of communities	3	3	7	7

TABLE I
CHARACTERISTICS OF EXPERIMENTAL DATASETS

The 95% confidence intervals are plotted. We first present results in terms of the raw metrics. The performance of the 3 different criteria are plotted against plain SimBet routing (denoted by "without IS"). Fig. 7 and 8 illustrate cases where ISNC brings some gain. However, the presence and amplitude of gain depends on the network parameters whose impacts are presented in a concise form in Fig. 11, commented later on. For the MIT trace, Fig. 7 shows that all 3 ISNC criteria allow to deliver a higher number of messages than "without IS". The average delay is higher with ISNC, because it is averaged over successful messages, hence, as ISNC manages to keep delivering more messages even for advanced time instants while "without IS" does not catch up in number of messages in late time instants, the delay average of the latter is lower. We observe that the number of hops can be lowered by ISNC: the packets that arrive late at their destination nodes in "without IS" travel through a higher number of hops. With ISNC, the number of hops is taken as the maximum between the number of hops of remedy and coded packet. Hence, a lower number of hops with ISNC means that the coded packet is relayed by less nodes than the uncoded packet in "without IS", which is allowed by the fact that ISNC eases the transmission of the coded packet (see Sec. V-A), specifically when the network gets more congested (i.e., for higher values of R or M). This is seen even more clearly in Fig. 11. The number of forwards is often lowered by ISNC. BM feature F1 can also account for that as more useless packets get erased in the right community ("without IS", they can be erased only by uncoded packets

destined to this community, with ISNC, they are also erased b
coded packets destined to this community, and mixing packe
together, the number of coded packets destined to a give
community can be higher than the number of uncoded packe
replaced by these coded packets).

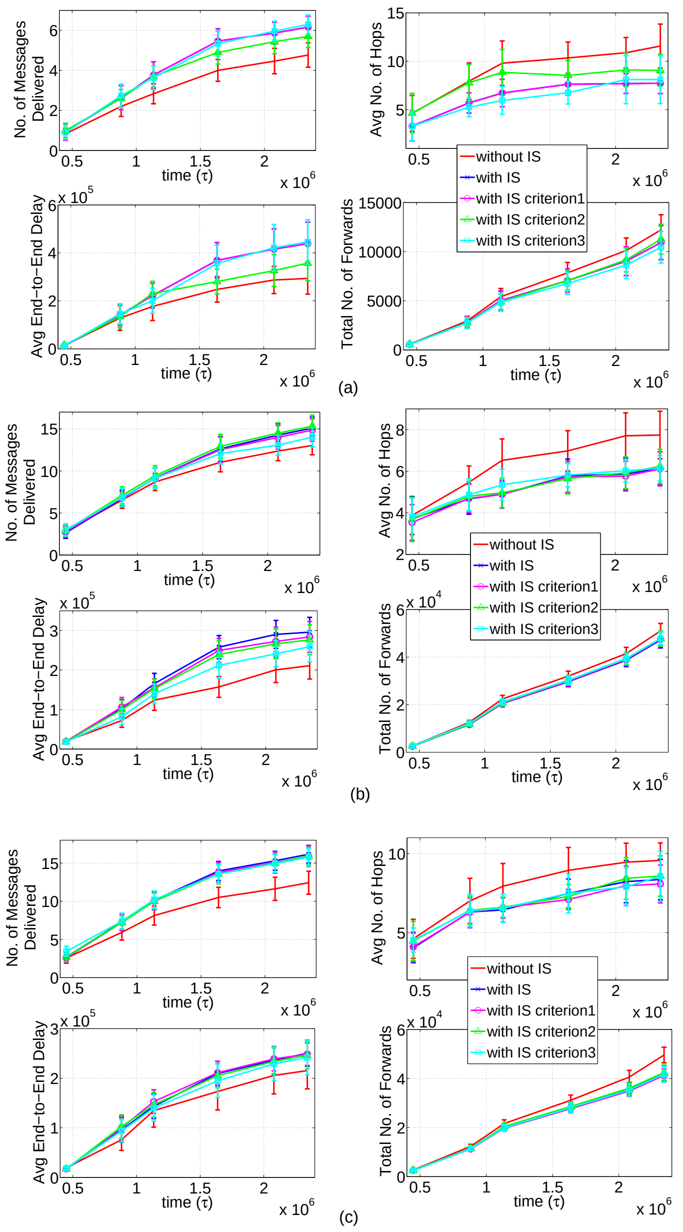

Fig. 7. MIT trace : (a) $R = 10$, $M = 40$, (b) $R = 50$, $M = 40$ and (c) $R = 50$, $M = 10$

We can see that the only case where there is a significant difference between the performance of the coding criteria is for MIT trace, $R = 10$ and $M = 40$. There, criterion 2 performs worse than 1 and 3, that is closer to “without IS”. If so, we can hypothesize the reason is that criterion 2 is not able to generate as many coded packets as the other two. We verify that in Fig. 9 which plots the number of coded packets over time. That means that criterion 2 is more stringent than the other two: criterion 1 considers a single hop as well, but a

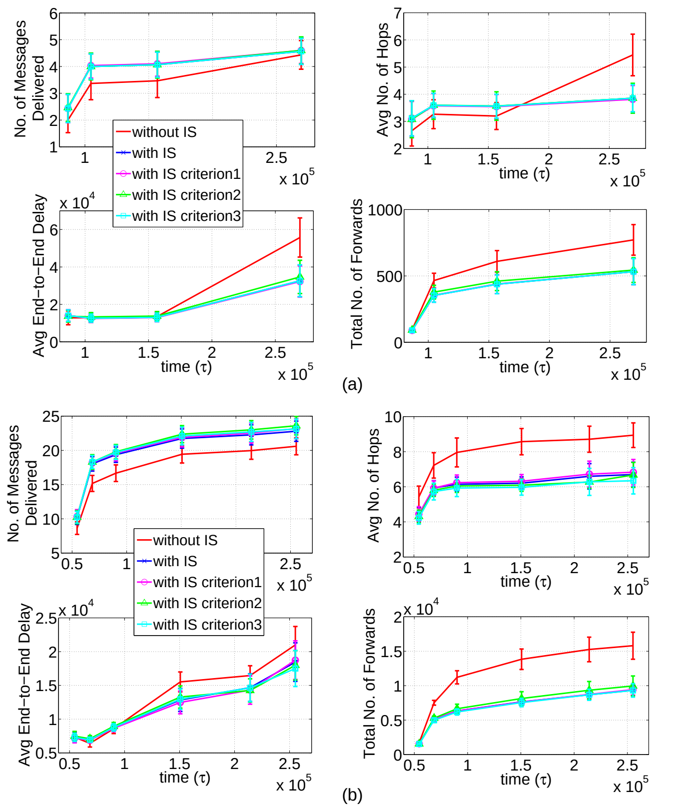

Fig. 8. (a) Intel trace: R=10, M=5, (b) Infocom05 trace: R=50, M=10

faster spreading where the utilities only impact the maximum infected nodes; criterion 2 considers the utilities the same way as criterion 3, but considers the set of one-hop and 2-hop paths for the remedy packets. This makes a difference in this setting where there are various paths between the different communities (see Fig. 10), a moderate number of sessions but M high enough so that the variety of the paths (for the remedies) can be exploited.

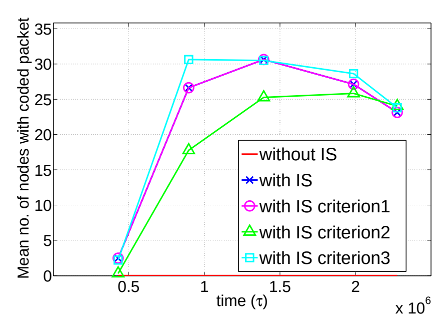

Fig. 9. Mean number of coded packets w.r.t time for MIT trace: R=10, M=40

Fig. 11 plots the coding gains for the four raw metrics. For the number of delivered messages, the higher the coding gain, the better ISNC, for the other metrics, the lower the coding gain, the better ISNC. The gains are plotted for fixed R and varying M (left column), and fixed M and varying R (right column). We present here results only for MIT and Intel. The reason is that MIT and Infocom exhibit the same

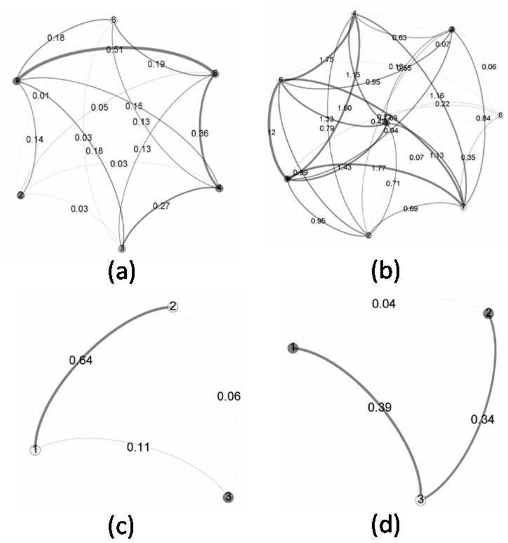

Fig. 10. Left: Mean number of coded packets w.r.t time for MIT trace: R=10, M=40. Right: The community structure of the traces (a) MIT, (b) Infocom05, (c) Intel and (d) Cambridge

kind of trends, probably because they have same order of participants (100 and 40) and communities (7), so do Intel and Cambridge (8 and 10 participants, 3 communities). We observe both in MIT and Intel that the gain in delivered messages exhibit a maximum in M and R. For higher M, the gain first increases then decreases with R. So there is an optimal level of congestion where ISNC helps most: for given R, M must be high enough to allow remedy packets to propagate, but not too high so that routing has faster propagation. The same observation can be drawn from the gain in delay. In terms of number of hops and number of forwards, for MIT, the higher M and R the better for the gain which then stabilizes. For Intel, for given R there seems to be an optimal M. We hypothesize that this difference between the traces is due to the low number of communities in Intel that prevents ISNC coupled with the proper BM to bring as high a gain as in MIT with numerous communities.

To conclude on the gain for these metrics, it seems that our criterion is able to grab opportunities to increase the number of delivered messages, which was our primary objective, while allowing gains on other metrics (like number of hops for high R) or containing the degradation that may be incurred on them, stabilizing it with M and R. For traces with very few communities, such as Intel, gains can be obtained for low M and R, but are more difficult to maintain (or equivalently, degradation to limit), for higher values. We think the reason is that the very design of both the criterion and the BM assumes various communities.

Finally, we illustrate the impact of ISNC on fairness in Fig. 12. The relative load is plotted for every mobile nodes ordered according to their betweenness (which is one of the SimBet utility components proper to a node and independent of destination, allowing to average). We observe that ISNC improves fairness as the relative load of high ranked ("highly popular" as in [35]) is decreased while that of less popular

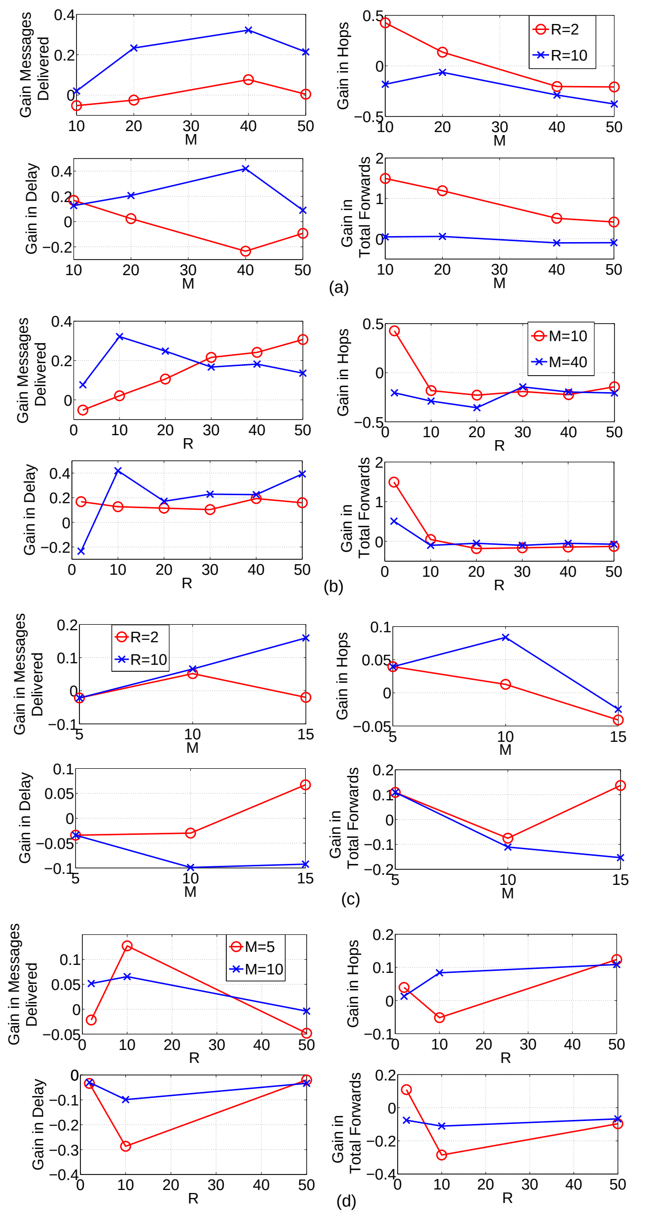

Fig. 11. Coding gains: (a) MIT: Gain vs. R, (b) MIT: Gain vs. M, (c) Intel: Gain vs. R, and (d) Intel: Gain vs. M

nodes is slightly increased. From the first three elements itemized in Sec. V-A, it can be seen that, during congestion, the routes for $P^{(1)}$ and $P^{(2)}$ are likely to get loaded more evenly by the coded packet than if only uncoded packets would be sent, as the probability that only one is scheduled, and hence only one is able to occupy its route while the other is hindered, is higher in the latter case. However, for other settings where ISNC is less beneficial for the other metrics too, the fairness can be worsened. Hence, though our coding criteria has not been designed with a fairness objective, it is interesting to note that it is another metric which can be positively impacted by ISNC, and hence can be envisioned as an objective to take into account to design possibly more efficient criteria.

VII. Conclusion

Despite DTMSNs have features (undirected and not time-shared) for which the current literature shows that ISNC may

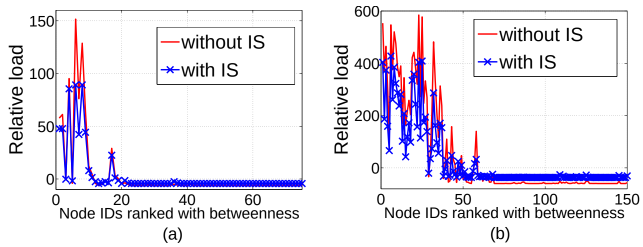

Fig. 12. Relative load per node: (a) Intel: R=10, M=5 and, (b) Infocom05: R=50, M=10

not be competitive to optimal routing, we have shown that ISNC is useful in handling several unicast sessions routed with social-aware routing which introduces a certain amount of directionality in the network. We discussed the closely tied interplay between ISNC and buffer management. We have devised heuristic decentralized ISNC criteria which trigger ISNC conservatively, taking into account the network load and the copy budget. We have extensively tested these criteria on real-world contact traces and shown the gains brought (or degradation contained) by ISNC on several metrics, and how they depend on the network parameters. We point out that these ISNC policies can be used on top of any social-aware routing policy. Future works include building on the results of [24] to express formally the ISNC optimization problem with the settings considered in the present article, to design ISNC policies as decentralized solutions to this problem. One could also consider the more general approach where the sources are allowed to purposely disseminate extra remedy packets, requiring to handle the optimization problem carefully to deal with the thereby incurred extra load.

References

[1] N. Shrestha and L. Sassatelli, “Inter-session network coding in delay tolerant mobile social networks: an empirical study,” in *IEEE Autonomic and Opportunistic Communications (AOC), WoWMoM workshop*, Boston, MA, USA, Jun. 2015.

[2] “Bytewalla DTN on Android Phones.” [Online]. Available: http://sourceforge.net/projects/bytewalla/

[3] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Pocket switched networks and human mobility in conference environments,” in *ACM SIGCOMM workshop on DTN (WDTN)*, Aug. 2005.

[4] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, “A socio-aware overlay for publish/subscribe communication in delay tolerant networks,” in *ACM MSWiM*, Chania, Greece, Oct. 2007.

[5] “Liberouter: Towards autonomous neighborhood networking.” [Online]. Available: http://www.ict-scampi.eu/results/scampi-liberouter/

[6] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, “Microcast: Cooperative video streaming on smartphones,” in *ACM MobiSys*, Jun. 2012.

[7] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay tolerant networks,” *IEEE Trans. on Mob. Comp.*, vol. 10, no. 11, pp. 1576 – 1589, Nov. 2011.

[8] E. M. Daly and M. Haahr, “Social network analysis for information flow in disconnected delay-tolerant MANET,” *IEEE Trans. on Mob. Comp.*, vol. 8, no. 5, pp. 606 – 621, May 2009.

[9] A. Mtibaa, M. May, C. Diot, and M. Ammar, “Peoplerank: Social opportunistic forwarding,” in *IEEE INFOCOM*, San Diego, USA, Mar. 2010.

[10] Y. Lin, B. Li, and B. Liang, “Efficient network coded data transmissions in disruption tolerant networks,” in *IEEE INFOCOM*, Phoenix, USA, Apr. 2008.

[11] R. Koetter and M. Médard, “An algebraic approach to network coding,” *IEEE/ACM Trans. on Netw.*, vol. 11, no. 5, 2003.

[12] A. Eryilmaz and D. S. Lun, “Control for inter-session network coding,” in *IEEE Inf. Th. and Appl. Workshop*, San Diego, USA, Jan. 2007.

[13] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, “Network coding for multiple unicasts: An approach based on linear optimization,” in *IEEE Int. Symp. on Inf. Th. (ISIT)*, Seattle, USA, Jun. 2006.

[14] A. Agarwal and M. Charikar, “On the advantage of network coding for improving network throughput,” in *IEEE Inf. Th. Workshop*, Oct. 2004, pp. 247–249.

[15] Z. Li and B. Li, “Network coding: the case of multiple unicast sessions,” in *Allerton Ann. Conf. on Comm., Contr., and Comp.*, 2004.

[16] K. Jain, V. V. Vazirani, and G. Yuval, “On the capacity of multiple unicast sessions in undirected graphs,” *IEEE/ACM Trans. on Inf. Th.*, vol. 52, no. 6, pp. 2805 – 2809, Jun. 2006.

[17] K. Cai and G. Han, “On network coding advantage for multiple unicast networks,” in *IEEE Int. Symp. on Inf. Th. (ISIT)*, Jun. 2015.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, “XORs in the air: Practical wireless network coding,” *IEEE/ACM Trans. on Netw.*, vol. 6, no. 3, 2008.

[19] A. Picu, T. Spyropoulos, and T. Hossmann, “An analysis of the information spreading delay in heterogeneous mobility DTNs,” in *IEEE WoWMoM*, San Francisco, CA, Jun. 2012.

[20] A. Picu and T. Spyropoulos, “DTN-Meteo: Forecasting the performance of DTN protocols under heterogeneous mobility,” *IEEE/ACM Trans. on Netw.*, vol. 23, no. 2, Apr. 2015.

[21] 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, no. 1, pp. 77–90, February 2008.

[22] S. Ahmed and S. S. Kanhere, “Hubcode: hub-based forwarding using network coding in delay tolerant networks,” *Wireless Comm. and Mob. Comp.*, vol. 13, no. 9, p. 828846, May 2011.

[23] X. Zhang, G. Neglia, J. Kurose, and D. Towsley, “Benefits of network coding for unicast applications in disruption-tolerant networks,” *IEEE/ACM Trans. on Netw.*, vol. 21, no. 5, pp. 1407–1420, Oct. 2013.

[24] N. Shrestha and L. Sassatelli, “On control of inter-session network coding in delay-tolerant mobile mocial networks,” in *ACM MSWiM*, Montreal, Canada, Sep. 2014.

[25] T. Karagiannis, J.-Y. L. Boudec, and M. Vojnović, “Power law and exponential decay of inter contact times between mobile devices,” in *ACM MobiCom*, Montreal, Canada, Sep. 2007.

[26] S. Maheshwar, Z. Li, and B. Li, “Bounding the coding advantage of combination network coding in undirected networks,” *IEEE Trans. on Info. Th.*, vol. 58, no. 2, pp. 570– 584, 2012.

[27] G. Alfano, M. Garetto, and E. Leonardi, “On the capacity region of MANET: Scheduling and routing strategy,” *IEEE Trans. on Veh. Tech.*, vol. 58, no. 4, pp. 1930–1941, 2009.

[28] A. Balasubramanian, B. Levine, and A. Venkataramani, “DTN routing as a resource allocation problem,” *ACM SIGCOMM Comp. Comm. Rev.*, vol. 37, no. 4, pp. 373–384, Oct. 2007.

[29] T. Spyropoulos, T. Turletti, and K. Obraczka, “Routing in delay-tolerant networks comprising heterogeneous node populations,” *IEEE Trans. on Mob. Comp.*, vol. 8, no. 8, pp. 1132 – 1147, Aug. 2009.

[30] M. Langberg and M. Médard, “On the multiple unicast network coding conjecture,” in *Allerton Ann. Conf. on Comm., Contr., and Comp.*, 2009.

[31] A. Krifa, C. Barakat, and T. Spyropoulos, “Message drop and scheduling in DTNs: Theory and practice,” *IEEE Trans. on Mob. Comp.*, vol. 11, no. 9, pp. 1470 – 1483, Sep. 2012.

[32] P. Hui, E. Yoneki, S. Chan, and J. Crowcroft, “Distributed community detection in delay tolerant networks,” in *ACM SIGCOMM Workshop (MOBIARCH)*, Kyoto, Japan, Aug. 2007.

[33] “A community resource for archiving wireless data at Dartmouth.” [Online]. Available: http://crawdad.cs.dartmouth.edu/index.php

[34] T. Hossmann, T. Spyropoulos, and F. Legendre, “Know thy neighbor: Towards optimal mapping of contacts to social graphs for DTN routing,” in *IEEE INFOCOM*, San Diego, USA, Mar. 2010.

[35] A. Mtibaa and K. A. Harras, “FOG: Fairness in mobile opportunistic networking,” in *IEEE SECON*, Salt Lake City, USA, 2011, pp. 260–268.

Appendix

A. Proof of Lemma 5.1

Consider the 2-community network. To compute the conductance of this graph, according to Eq. (3), we need to find the cut (set of nodes) leading to the minimum $\phi(S)$. Owing to the per-community homogeneity of the nodes, a cut in such a network is defined by α_1 and α_2 the number of infected nodes in community 1 and 2, respectively. Let α be $\alpha_1 + \alpha_2$. With the definition provided in Sec. V-A, we have

$$\partial(S_{\alpha,\alpha_1}) = f(\alpha,\alpha_1) = \frac{\alpha_1 p_{11} + (\alpha-\alpha_1)p_{22} + \alpha p_{12}}{\alpha(N_{eff}-\alpha)}\ .$$

We can easily work out the minimization of this two-variate function $f(\alpha,\alpha_1)$. The derivative with respect to α_1 is $\frac{\partial f}{\partial \alpha_1} = \frac{p_{11}-p_{22}}{\alpha(N_{eff}-\alpha)}$.

- If $p_{11} - p_{22} \geq 0$, the minimum is obtained for $\alpha_1 = 0$ then for $\alpha = 1$, and we get $\Phi = \frac{p_{22}+p_{12}}{N-1}$.
- If $p_{11} - p_{22} < 0$, the minimum is obtained for α_1 maximum, that is $\alpha_1 = \alpha$ then for $\alpha = 1$, and we get $\Phi = \frac{p_{11}+p_{12}}{N-1}$.

Finally, coming back to the continuous time scale: $\Phi = \min(\beta_{11}, \beta_{22}) + \beta_{12}$. ⋄

B. Proof of Lemma 5.2

The conductance $\Phi^{(c)}$ is that of a 3-community network. The derivation of $\Phi^{(c)}$ unfolds as that of Φ for criterion 1, except we deal with α_1, α_2 and α_c for the three communities, or equivalently with $\alpha = \alpha_1 + \alpha_2 + \alpha_c$, α_1 and α_2. We hence have

$$\Phi^{(c)} = \min_{\alpha,\alpha_1,\alpha_2} f(\alpha,\alpha_1,\alpha_2) =$$

$$\min_{\alpha,\alpha_1,\alpha_2} \frac{\alpha_1(p_{11}+p_{12}+p_{1c}) + \alpha_2(p_{22}+p_{12}+p_{2c}) + \alpha_c(p_{cc}+p_{1c}+p_{2c})}{\alpha(N_{eff}-\alpha)}\ .$$

Let A, B and C temporarily denote the 3 factors in brackets in the above numerator, $D = A - C$ and $E = B - C$. The minimization then writes as follows. We have $\frac{\partial f}{\partial \alpha_1} = \frac{D}{\alpha(N_{eff}-\alpha)}$

- If $D \geq 0$, the minimum is obtained for $\alpha_1 = 0$ then $f(\alpha, 0, \alpha_2) = \frac{\alpha_2 E + \alpha C}{\alpha(N_{eff}-\alpha)}$
  - if $E \geq 0$, the minimum is obtained for $\alpha_2 = 0$ then $\frac{\alpha C}{\alpha(N_{eff}-\alpha)}$ is minimum for $\alpha = 1$, whereby $\Phi = \frac{C}{(N_{eff}-1)}$
  - if $E < 0$, the minimum is obtained for $\alpha_2 = \alpha$ then for the same reasons as above: $\Phi = \frac{E+C}{(N_{eff}-1)}$
- If $D < 0$, the minimum is obtained for $\alpha_1 = \alpha$ then $f(\alpha,\alpha,\alpha_2) = \frac{\alpha_2 E + \alpha(D+C)}{\alpha(N_{eff}-\alpha)}$
  - if $E \geq 0$, the minimum is again obtained for $\alpha_2 = 0$ and $\alpha = 1$, whereby $\Phi = \frac{D+C}{(N_{eff}-1)}$
  - if $E < 0$, the minimum is obtained for $\alpha_2 = \alpha$ and $\alpha = 1$, whereby $\Phi = \frac{\min(D,E)+C}{(N_{eff}-1)}$

We therefore obtain the expression in Eq. (4) to take into account all possible above cases. ⋄

C. Detailed ISNC protocol for relay-relay exchange

Algorithm 1: Protocol with Multi-session ISNC for Relay-Relay transfer

Data: Node A in community attempting to transfer to node B, each in community a and b with l^A and l^B packets, respectively. The number of different sessions N_S, buffer size for each node **B**, each session i has source-destination pair as (S_i, D_i), $S = \{S_1, \ldots, S_{N_S}\}$, $D = \{D_1, \ldots, D_{N_S}\}$. Index i describes the packet index (session) of A in decreasing order of copy counter. Total number of packets at A is I. The current packet at A (resp. B) is denoted p (resp. q). Below we use:
$p.counter(SimBet) = \left\lfloor p.counter \frac{U_B(D_i)}{U_A(D_i)+U_B(D_i)} \right\rfloor$ if p is not a coded packet,
$p.counter(SimBet) = \left\lfloor p.counter \left(\frac{U_B(D_c)}{U_A(D_c)+U_B(D_c)} + \frac{U_B(D_d)}{U_A(D_d)+U_B(D_d)} \right) \right\rfloor$ if p is packet coding sessions c and d.

Result: Field values of the packet structure at A and B after transfers

```
i = 1;
while A and B in radio range and i ≤ I do
    if A ≠ S_i and B ≠ D_i then
        if B − l^B ≥ 1 and p.index not in B then
            Packet q is created at B with:
            (i) q.index = p.index; (ii)
            q.counter = p.counter(SimBet); (iii)
            q.payload = p.payload;
            Update
            (i) l^B = l^B + 1; (ii)
            p.counter = p.counter − p.counter(SimBet);

        else if it exists, pick q with the lowest counter such
        that p satisfies F1 with q then
            Overwrite q with p and
            q.counter = q.counter + p.counter(SimBet);
            Update:
            p.counter = p.counter − p.counter(SimBet);
        else if it exists, pick q with the lowest counter such
        that p satisfies F2 with q then
            Only update
            q.counter = q.counter + p.counter(SimBet)
            and
            p.counter = p.counter − p.counter(SimBet);
        else if it exists, pick q with the greatest sum of
        differences between left-hand and right-hand sides in
        inequality 1 and such that p satisfies the coding
        criterion with q then
            Let c = p.index and d = q.index. Replace q with
            (i) q.index = N_s + c + d − 1; (ii)
            q.counter = p.counter(SimBet) + q.counter;
            (iii)
            q.payload = RLC(p.payload and q.payload);
            Update:
            p.counter = p.counter − p.counter(SimBet);

end
```

Go to the beginning of Algo. 1, exchange A and B and perform again all the steps for packets not exchanged yet.

Note: F4 and exchanges with sources and destinations are left out intentionally for the sake of clarity.

Neetya Shrestha obtained her MSc jointly from Asian Institute of Technology, Bangkok, Thailand, and Telecom SudParis, France, in 2011. She obtained the PhD degree from Université Nice Sophia Antipolis, France, in 2015. Her research interests have focused on delay tolerant networks, network coding and P2P systems. She is now with the Amadeus company.

Lucile Sassatelli has been an assistant professor at Université Nice Sophia Antipolis, France, since 2009. She received the M.Sc. and the Ph.D. degrees in telecommunication engineering from the University of Cergy-Pontoise, France, in 2005 and 2008, respectively. In 2008-2009, she was a postdoctoral fellow in the Reliable Communications and Network Coding Group (RLE) at Massachusetts Institute of Technology (MIT), Cambridge, USA. Her research activities focus on network coding, disruption tolerant networks and mobile social networks, video streaming and optimization.