Improving the transport performance in delay tolerant networks by random linear network coding and global acknowledgments

Arshad Ali, Manoj Panda, Tijani Chahed, Eitan Altman

▶ **To cite this version:**

Arshad Ali, Manoj Panda, Tijani Chahed, Eitan Altman. Improving the transport performance in delay tolerant networks by random linear network coding and global acknowledgments. Ad Hoc Networks, 2013, 11 (8), pp.2567-2587. 10.1016/j.adhoc.2013.07.004 . hal-00916916

HAL Id: hal-00916916

https://inria.hal.science/hal-00916916

Submitted on 10 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*Manuscript

Improving the Transport Performance in Delay Tolerant Networks by Random Linear Network Coding and Global Acknowledgments

Arshad Ali[a], Manoj Panda[b,*], Tijani Chahed[c], Eitan Altman[d]

[a] *Orange Labs, 92130 Issy les Moulineaux, France.*
[b] *Swinburne University of Technology, John Street, Hawthorn, 3122 Victoria, Australia.*
[c] *Telecom SudParis, 9 rue C. Fourier, 91011 Evry Cedex, France.*
[d] *INRIA, 2004 Route des Lucioles, 06902 Sophia-Antipolis, France.*

Abstract

We propose and study a new set of enhancement features to improve the performance of reliable transport in Delay Tolerant Networks (DTNs) consisting of both unicast and multicast flows. The improvement in reliability is brought in by a novel Global Selective ACKnowledgment (G-SACK) scheme and random linear network coding. The motivation for using network coding and G-SACKs comes from the observation that one should take the maximum advantage of the contact opportunities which occur quite infrequently in DTNs. Network coding and G-SACKs perform "mixing" of packet and acknowledgment information, respectively, at the contact opportunities and essentially solve the *randomness* and *finite capacity* limitations of DTNs. In contrast to earlier work on network coding in DTNs, we observe and explain the gains due to network coding even under an *inter-session* setting. Our results from extensive simulations of appropriately chosen "minimal" topologies quantify the gains due to each enhancement feature. We show that substantial gains can be achieved by our proposed enhancements that are very simple to implement.

Keywords: Delay Tolerant Networks (DTNs); reliable transport; multicast; inter-session network coding; delay reduction, sample mean and variance; pairwise delay; network-wide maximum delay

1. Introduction

Mobile Ad hoc NETworks (MANETs) aim at making communication between mobile nodes feasible without any infrastructure support. If the spatial density of mobile nodes in a MANET is too low, then an end-to-end path between a source and a destination almost never exists, and two mobile nodes can communicate only when they come within the radio range of each other. Such sparse and/or highly mobile MANETs fall into the class of Delay Tolerant Networks (DTNs) that are characterized by frequent link disruptions and highly intermittent connectivity. There can be several reasons for intermittent connectivity such as limited wireless radio range, sparsity of mobile nodes, limited energy resources, attack, and noise [2].

Apart from sparse MANETs, other examples of DTNs include Inter-Planetary Networks (IPNs) [3] which pertain to *deep-space* communication. Examples of *terrestrial* applications of DTNs include sparse Vehicular Ad hoc NETworks (VANETs) [4], Pocket Switched Networks (PSNs) [5], Airborne Networks (ANs) [6], Mobile Social Networks (MSNs) [7], Under Water Networks (UWNs) [8] and "Data Mules" [9].

In DTNs, due to highly intermittent connectivity, no contemporaneous end-to-end path may ever exist [10], and hence, the nodes must adopt a *Store-Carry-and-Forward* paradigm of routing. A source has to depend on the mobility of other nodes, which act as "relays", and data packets and ACKnowledgments (ACKs) get transferred between a source and a destination through one or more relays. This in-

⋆This paper is a thoroughly revised and substantially extended version of our earlier work [1].

⋆⋆The research work was carried out when the first two authors were affiliated with Telecom SudParis.

*Corresponding Author

Email addresses: arshad.ali@orange.com (Arshad Ali), mpanda@swin.edu.au (Manoj Panda), tijani.chahed@it-sudparis.eu (Tijani Chahed), eitan.altman@sophia.inria.fr (Eitan Altman)

herently entails a large delay. Since the nodes take advantage of the transmission opportunities during contacts with other nodes in order to exchange packets, such forwarding mechanism is sometimes also referred to as *opportunistic routing*.

In reliable transport, a source wishes to ensure that the information it sends arrive correctly and "in order" at the destination. The Transmission Control Protocol (TCP) is by far the most deployed protocol for reliable transport. However, *TCP turns out to be very inefficient for reliable transport in MANETs, because it misinterprets losses due to interference and link failures as losses due to congestion* [11]. This is even worse in the case of DTNs due to intermittent connectivity [12]. Transport solutions for MANETs, that are based on cross-layer signaling [13], [14], [15], are not suitable for DTNs, because only opportunistic routing can be performed.

Clearly, the performance of routing and transport in DTNs is very much dependent both on the *mobility of the nodes* and the *packet replication method* [16], and there is a need for transport solutions that could leverage this special characteristics of DTNs. In this paper, we propose and study several "smart" techniques for replicating packets and acknowledgments in order to improve the performance of reliable transport in DTNs, and show their efficacy under a realistic heterogeneous mobility model as well as a homogeneous mobility model.

1.1. Literature Survey

Several methods for spreading packets in DTNs have been investigated under opportunistic routing, for example, *spray-and-wait routing* [17], *probabilistic routing* [18], *direct delivery* and *two-hop routing* [19], and *epidemic routing* [20]. Much of the existing literature on DTNs focuses on the routing aspect and relatively fewer pieces of work deal with reliable transport. The literature on transport in DTNs is primarily concerned with deep-space communication [21], [22], [23], [24], [25], [26], [27].

The Bundle Protocol [21] specifies a framework rather than a concrete protocol implementation. The "Saratoga" protocol [22] provides an IP-based convergence layer in DTNs supporting store-and-forward of bundles. It performs UDP-based transfer of IP packets with Selective Negative ACKnowledgements (SNACKs). The Licklider Transmission Protocol (LTP) [23] is designed to serve as a DTN convergence layer protocol. It provides retransmission-based reliable transfers over single-hop connections and supports both reliable and unreliable data transmission. The CFDP protocol [24] provides file copy services over a single link and requires all parts of a file to follow the same path to the destination. The Deep-Space Transport Protocol (DS-TP) [25] is based on *Double Automatic Retransmission* to provide proactive protection against link errors. The TP-Planet [26] protocol employs *Additive Increase Multiplicative Decrease* control mechanism and uses time-delayed Selective ACKnowledgments (SACKs) to deal with asymmetric bandwidth. SCPS-TP [27] adopts major TCP functionalities and extends them in order to deal with some of the unique characteristics of deep-space links. Harras and Almeroth [28] probed into issues related to the use of transport in a DTN environment. Fall and MaCanne [29] discussed important issues related to transport performance not specific to DTNs.

Ahlswede et al [30] initiated the study of network coding. Ho et al [31] proposed the random linear coding technique. Network coding research originally studied throughput performance without delay considerations for channels with no erasures and no feedback [30, 32, 33]. The network coding based routing approaches for challenging environments such as DTNs have been widely studied and simulated, and have been shown to provide promising results [34, 35, 36, 37, 38]. Widmer and Le Boudec [37] show that the performance of their network coding based routing algorithm is better than that of probabilistic routing. Katti et al [34] showed that network coding can improve the throughput in unicast wireless communication. Zhang et al [38] investigated the benefits of using random linear coding with epidemic routing for unicast communications in mobile DTNs, especially *under constrained buffer sizes and intra-session coding*. Network coding has been used not only to minimize the delivery delay [39], [38] but also to improve the probability of successful delivery within a given time [40].

In contrast to the large body of work on transport in deep-space DTNs, in [41] the authors proposed a new reliable transport protocol for "terrestrial" DTNs. The reliable transport scheme in [41] is based on random linear coding of data packets and uses a special type of ACKs that indicate the *missing degrees-of-freedom (DoF)* at the destination. The scheme in [41] operates in (re)transmission cycles. If the source does not receive an ACK indicating zero missing DoF within an optimally chosen

cycle timeout, then a new (re)transmission cycle, with updated optimal parameter settings based on the latest information of missing DoF, is triggered. This process continues until the source receives an ACK indicating zero missing DoF, and thus, reliability is achieved.

1.2. Our Contributions

This work is an extension of our earlier work [1] in which we proposed several enhancement features to improve the performance of reliable transport in terrestrial DTNs. In this work, we extend the conclusions of [1] by quantifying the benefits due to each enhancement feature under appropriately chosen "minimal" topologies. We study the performance benefits using a wider range of performance metrics and provide more insightful observations through critical analysis of the simulation results.

Since designing appropriate ACK mechanisms in DTNs is still an open issue [42], we propose and study several enhanced ACK schemes. We propose a novel Global Selective ACKnowledgement (G-SACK) scheme. A G-SACK can potentially contain *global* information about the receipt of packets at each destination in the network. We also study the impact of our enhanced ACK schemes together with random linear network coding.

Our overall proposed scheme, which combines random linear network coding of packets and G-SACKs, provides an improvement, with homogeneous (resp. heterogeneous) mobility, up to 26.1%, 82.1% and 75.4% (resp. 150.6%, 78.4% and 56.2%) in the network-wide round trip success probability, and the mean and standard deviation of the network-wide maximum round trip delay, respectively, as compared to a baseline scheme. Our main insight is that this substantial improvement comes from the *mixing of information inside the network* which essentially solves the two primary limitations of DTNs, namely, *randomness* and *finite capacity*.

To the best of our knowledge, this is the first work which explores the benefits of random linear combinations of packets from different sources (*inter-session* network coding) in a DTN with epidemic-type routing. These benefits include improvement in reliability (enhancing the probability of successful delivery of packets) and minimization of transfer delay for unicast as well as for multicast flows. We show that random linear network coding *makes the forward path smoother*, i.e., it decreases delay variance, and *improves the fairness among flows*.

The enhancement schemes in this paper should be viewed as improving the performance within each (re)transmission cycle of a reliable transport scheme, such as in [41]. The enhancement features in this paper increase the probability of receiving the final ACK (i.e., the ACK indicating zero missing DoF) within a given timeout. Equivalently, they decrease the waiting times (i.e., timeouts) required to achieve a given probability of receiving the final ACK. A sequence of (re)transmission cycles is still needed to achieve reliability, i.e., the enhancement schemes in this paper do not eliminate the need for (re)transmission cycles, but they reduce the mean number of cycles as well as the mean cycle timeouts.

1.3. Organization of the Paper

In Section 2, we describe our network setting and the mobility model. Section 3 discusses the issues that motivate our enhancements. In Section 4, we propose our enhancements. In Section 5, we provide the simulation settings and define the performance metrics. Section 6 and Section 7 exhibit the gains brought by each enhancement feature. In Section 8, we show benefits of our overall proposed scheme. Section 9 concludes the paper.

2. Network Setting

We consider a DTN consisting of $S + D + N$ *mobile* nodes. There are S *source* nodes, D *destination* nodes and N *relay* nodes (see Figure 1). We allow for both *unicast* (one source to one destination) and *multicast* (one source to multiple destinations) data transfer. Also, a node can be a destination for multiple sources. The sources send packets to their destinations through the relays. The destinations send back acknowledgments, for every received packet, to their corresponding sources through the relays. We study packet transfer with and without network coding of the payload parts, and with three different acknowledgment schemes – a simple *baseline* scheme and two enhancements to it (see Section 4). Packets and acknowledgments are transferred through the relays as in epidemic routing [20] with slight modifications (the details of which will be provided later).

Each source-destination pair (i, j), $i = 1, \ldots, S$, $j = 1, \ldots, D$, defines a *flow*. A multicast flow $(i, \mathcal{J})$ consists of multiple flows with the same source and multiple (different) destinations, i.e., for all $\mathcal{J} \subset$

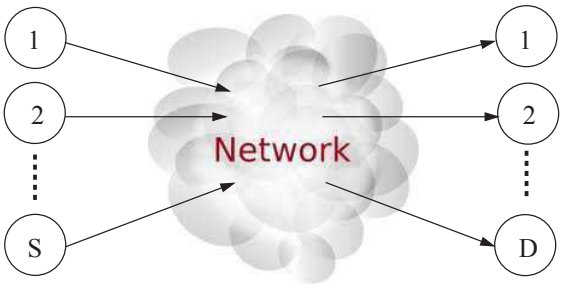

Figure 1: The network of N mobile relays (shown as the cloud) connecting the S sources and the D destinations. The sources and destinations are also mobile.

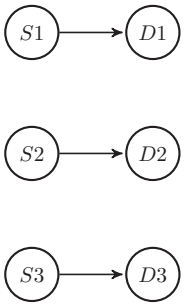

Figure 2: Topology 3: three unicast flows. (Other topologies are discussed later)

$\{1, \ldots, D\}$ such that $|\mathcal{J}| \geq 2$, we define

$$(i, \mathcal{J}) := \{(i, j') : j' \in \mathcal{J}\}.$$

The *flow matrix* $A = [a_{ij}]$ is an $S \times D$ matrix, where, for all i and j, $i = 1, \ldots, S$, $j = 1, \ldots, D$, the entry $a_{ij} = 1$, if source i has a packet to send to destination j; otherwise, $a_{ij} = 0$. For example, the flow matrix corresponding to the network in Figure 2, which consists of three unicast flows, is given by

$$A_U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

We denote the number of (pairwise) flows by N_{pw}. Clearly, $N_{pw} = \sum_{i=1}^{S} \sum_{j=1}^{D} a_{ij}$.

Mobility Model: Two nodes are said to "meet" when they come within the *communication range* of each other. The key quantities that characterize the mobility pattern and significantly impact the performance in opportunistic networks are the *inter-meeting times of node pairs* [43], [44].

In [43], the authors thoroughly examine real-world mobility traces collected in several different network scenarios such as WiFi, vehicular GPS, GSM and Bluetooth, and conclude that the Complementary Cumulative Distribution Function (CCDF) of the *aggregated* inter-meeting time (over all pairs of nodes in the network) follows a *power law with exponential cutoff*. In [44], the authors show that the above "aggregate" inter-meeting time distribution would result when the inter-meeting time of each "individual" pair is exponentially distributed, but different node pairs have different rate parameters sampled from a Pareto distribution. In [45, 46, 47], the authors provide simulation results to show that, for *random waypoint* mobility models such as the *random direction* and *random walk* mobility models, one can accurately approximate the actual inter-meeting times by assuming the inter-meeting times for all node pairs to be i.i.d. exponential random variables.

Following [43] and [44], we consider a *heterogeneous* mobility model where the inter-meeting time of each "individual pair" is exponentially distributed, but different node pairs have different rate parameters sampled from a Pareto distribution given by the Cumulative Distribution Function (CDF)

$$F(x) = 1 - \left(\frac{x_{min}}{x}\right)^{\alpha},$$

where α is called the *shape* parameter and x_{min} is called the *scale* parameter. As shown in [44], the CCDF of the resulting aggregate inter-meeting time follows a *power law with exponential cutoff* which is exhibited by real-world mobility traces [43]. Following [45, 46, 47], we also consider a *homogeneous* mobility model where the inter-meeting times of the individual node pairs are i.i.d. exponentially distributed random variables with a rate parameter β.

To be able to compare the results obtained with homogeneous and heterogeneous mobility models, we chose the parameters of the heterogeneous mobility model in such a way that the network-wide expected meeting rate in the case of heterogeneous mobility is equal to the rate β in the homogeneous mobility model. In particular, we first set $\alpha = 1.5$,[1]

[1]Most real-world occurrences of Pareto distribution satisfy $1 < \alpha < 2$, which implies a finite first moment and infinite second and higher moments [48]. In our case, $\alpha = 1.5$ which implies that the mean rate is finite, but the sample variance of the rate parameter (over all node pairs) would increase with the increase in the number of nodes. In particular, the sample variance of the rate parameter would be infinite if there were infinite number of nodes.

and then equate the mean rate $\bar{x} = \frac{\alpha}{\alpha-1}x_{min}$ with β, and finally solve to obtain $x_{min} = \beta/3$.

Link Quality During Contacts: The time duration for which two nodes remain within each other's communication range is called the *contact duration*. As a consequence of the varying distance between the mobile nodes and fading effects, the link quality between two nodes changes over time during a contact. The amount of data that can be exchanged in a meeting is determined by the contact duration as well as the link quality during a contact. To form a network coded packet, each of the two participating nodes should be able to successfully receive a packet from the other node. In general, the number of packets that can be exchanged between two nodes during a contact is random. We assume that *whenever two nodes meet the contact duration is long enough for each node to successfully receive a packet from the other node and form a network coded packet.*

3. Fundamental Limitations of DTNs

Let flow (i, j), $i = 1, \ldots, S$, $j = 1, \ldots, D$, consists of N_{ij} packets. The unicast flow (i, j) is said to be *complete* when the acknowledgments for all the N_{ij} packets reach source i. A multicast flow $(i, \mathcal{J})$ is said to be complete when all the constituent flows (i, j'), $j' \in \mathcal{J}$, are complete.

From the performance perspective, it is desirable that (i) each flow is complete with as small a delay as possible, and (ii) there is fairness across the (pairwise) flows with respect to *throughput* (i.e., number of packets transferred per unit time). If the packet buffering and scheduling policy at the relays treat all flows equally, then one can expect to achieve *long-term* fairness (i.e., over a sufficiently large number of packet transfers for each flow). However, it is desirable to achieve *short-term* fairness as well (i.e., over a few packet transfers).

There are fundamental limitations that need to be addressed before one could achieve the desirable properties. The two primary limitations are *randomness* and *finite capacity*. Randomness refers to the randomness in the meeting/contact process pertaining to the mobility of the nodes. The *inter-meeting times* (i.e., the time duration between successive meetings of two nodes) and the *contact durations* (i.e., the time duration for which two nodes remain within each other's communication range) are random. Finite capacity refers to a finite number of relays with finite buffer space and the finiteness of the contact durations. Note that *the amount of data that can be exchanged in a meeting is determined by the contact duration.*

The round-trip delay associated with the transfer of a packet consists of a *forward* component (the delay after which the destination(s) receive the packet) and a *return* component (the delay after which the source receives the corresponding acknowledgment(s)). As the capacity (i.e., the number of relays and/or the buffer space at the relays and/or the contact duration) decreases and/or the mean inter-meeting time increases, the expected values of the forward, the return, and hence, the round-trip delay, increases. The randomness in the meeting/contact process, combined with finite capacity, determines the second and higher-order moments of the delays. Randomness makes the network inefficient as follows:

Inefficiency: With little or no information about the contact patterns among the nodes, a significant fraction (or percentage) of transmissions gets wasted in forwarding copies of packets (resp. acknowledgments) to those relays that do not contribute toward delivering the packets (resp. the acknowledgments) to the intended destinations (resp. sources).[2] Similarly, of the meetings a destination (resp. source) has with a relay, a significant fraction gets wasted because the packets (resp. the acknowledgments) carried by the relay are not intended for the destination (resp. source).

More importantly, randomness, combined with finite capacity, degrade the performance as follows:

Non-reachability: In reality, a source can wait for an acknowledgment only for a finite time T, and, with positive probability, does not receive the acknowledgment within T. We call this *Type-I* non-reachability. Due to randomness and finite capacity, with multiple competing flows, a source may not be able to transfer any of its packets even if it waits for infinite time. With positive probability, a source may never meet with a relay having

[2]The order in which relays get copies of a packet from a source can be modeled as a "tree". Starting with the source as the root, the tree can be "grown" over time as follows. Whenever a relay having the packet meets with a relay that does not have the packet but has the required buffer space to get a copy of the packet, a new leaf node is added by a link to the existing node which gives the packet. The first copy of the packet follows a unique path from the source to the destination on this tree, and the links on this path are "contributing". All links in the tree that do not belong to the path taken by the first copy of the packet from the source to the destination are "non-contributing".

the required amount of buffer space; indeed, due to the randomness of node meetings, the buffer space of every relay that meets with a source might already be completely occupied with packet(s) and acknowledgment(s) of other sources and destinations. We call this *Type-II* non-reachability, which can be solved by expunging the packet(s) and acknowledgment(s) from the relay buffers using (random or deterministic) *expiry timeouts* so that packets from all sources get access to relay buffers sooner or later. However, if the mean time to expiry is smaller than the mean time between relay meetings, then, with positive probability, the packet(s) will not reach the destination(s).

Inter-Dependence: A less severe but more likely problem is that of inter-dependence among the packet transfers, which occurs because they share the same resources (relay buffers and transmission times during contacts). Due to randomness and finite capacity, *in a particular realization of competing packet transfer processes*, one particular packet might get replicated at a much faster rate than others, leaving less resources for others. Due to the epidemic-type replication, small differences during the initial phase (which is unavoidable due to the randomness of node meetings) can become significant over time, since a packet with a larger number of copies replicates at a faster rate. Also, acknowledgments for a fast-spreading packet are generated before the other packets could reach their destinations. This implies that acknowledgments for a fast-spreading packet compete with slow-spreading packets for getting replicated. Similarly, the acknowledgments of different packets also compete for getting access to the finite relay buffers and for getting forwarded within the finite contact durations.

Thus, *one packet transfer being faster automatically implies that other packet transfers being slower*. This inequity results in large delay variance and short-term unfairness.

In Section 4, we propose enhancement schemes to address the issues identified in this section.

4. Our Proposed Enhancements

In Section 3 we identified that, with little or no information about the contact patterns among the nodes, randomness makes the network inefficient. Furthermore, there is competition (a) between packets, (b) between acknowledgments, and (c) between packets and acknowledgments, of the same or different flows for getting access to the finite relay buffers and getting replicated within the finite contact durations. This results in longer expected delays, larger delay variances and short-term unfairness.

The obvious solutions to the above problems are to: (1) increase the buffer capacity of the relays, (2) increase the transmit powers and/or apply sophisticated physical layer techniques (modulation/coding schemes) to increase the communication range, thereby, make the contact durations longer, (3) learn the contact patterns of the node meetings (which requires more processing and involves a "learning delay"), and (4) devise appropriate buffer management and scheduling policies for (i) admitting new packets and acknowledgments from other nodes, (ii) transmitting buffered packets and acknowledgments to other nodes, and (iii) expunging buffered packets and acknowledgments using expiry timeouts.

Our goal in this work, however, is to improve the performance with the limited available resource, with existing physical layer implementation, without using any information about the contact pattern, and over time-scales *finer* than the expiry timeouts. We propose and study our enhancements without referring to any buffer management and scheduling policy, since the study of buffer management and scheduling policies is out of the scope of this work. Interested readers may refer to [49, 50].

We avoid addressing the issue of buffer management and scheduling by restricting our study to ***one*** *packet per source (i.e.,* $N_{ij} = 1$ *for all* i, j*), and relay buffers with capacity* B *to store at the most* ***one*** *packet of length* L*,* $L \leq B < 2L$*, or a few acknowledgments that can fit into the buffer capacity* B*.* We also assume that the contact durations are long enough so that the relays can exchange all the information stored in their respective buffers with one another in a single meeting.

Our enhancements are detailed as follows:

(a) Generation of Selective ACKnowledgments (SACKs) at the destination: Upon receipt of a packet, a destination generates a Selective ACKnowledgment (SACK) indicating the set of sources from which it has already received the packet(s). This is in contrast with the baseline acknowledgment scheme, henceforth called "the (plain) ACK Scheme", in which only the currently received packet is acknowledged. A SACK can acknowledge *multiple sources* about the receipt of

packets at a *specific destination*.[3] Of course, if a node is a destination for a single source, then it can only generate an ACK.

(b) Update of ACKs/SACKs inside the network to form Global-ACKs/SACKs: When a node carrying an ACK (or a SACK) generated by a destination, on its way back to the source(s), meets with other nodes carrying ACKs/SACKs generated by other destinations (or more recent SACKs generated by the same destination), the information contained in the ACKs/SACKs are combined to form Global SACKs (G-SACKs). In the special case where all destinations generate only ACKs, we call the combined packet receipt information as Global ACKs (G-ACKs).

A G-SACK (resp. G-ACK) scheme is in contrast with the SACK (resp. the ACK) scheme in which the SACKs (resp. ACKs) generated by the destination(s) reach the sources without being updated inside the network. A G-ACK/G-SACK can acknowledge *multiple sources* about the receipt of packets at *multiple destinations*.

(c) Random linear network coding: Packets are combined at the relays to form random linear combinations. This allows packets from different sources to share the "packet payload space".

The key ideas that motivate our proposed enhancements are the following:

(I) The competition between packets of different flows can be mitigated by random linear coding of packets at the relays. A coded packet is more likely to contain useful information for each destination.

(II) The competition between acknowledgments of different flows can also be mitigated by coding of acknowledgments at the relays. However, since acknowledgments are much smaller than packets, we can combine the acknowledgment information, in uncoded form, efficiently as a matrix (discussed in detail below). This allows us to store the packet receipt information generated by a larger number of destinations within the same available relay buffer. Thus, a G-SACK is more likely to contain useful information for each source.

In the following, we elaborate on the implementation and other aspects. First, we define the *source-degree* and *destination-degree* as follows:

[3]Note that our notion of SACK is slightly different from the traditional one. In our case, SACK(s) from a destination provide information about receipt of packets from "different" sources and not different packets from the same source.

- **Source-degree:** The source-degree of a destination is the number of sources for which it is a destination. For example, the source-degree d_j^S of destination j is given by $d_j^S = \sum_{i=1}^{S} a_{ij}$. The average source-degree d^S in the network is defined by

$$d^S = \frac{\sum_{j=1}^{D} d_j^S}{D} = \frac{\sum_{j=1}^{D}\sum_{i=1}^{S} a_{ij}}{D} = \frac{N_{pw}}{D}.$$

- **Destination-degree:** The destination-degree of a source is defined as the number of destinations for which it is a source. For example, the destination-degree d_i^D of source i is given by $d_i^D = \sum_{j=1}^{D} a_{ij}$. The average destination-degree d^D in the network is defined by

$$d^D = \frac{\sum_{i=1}^{S} d_i^D}{S} = \frac{\sum_{i=1}^{S}\sum_{j=1}^{D} a_{ij}}{S} = \frac{N_{pw}}{S}.$$

It turns out that the benefits of our enhancements increase with increase in the source degree and/or the destination degree (see Section 8). Higher values of source and destination degrees also enable us to implement the G-SACKs in an efficient manner as matrices.

The G-SACK as a matrix: An ACK consists of the 3-tuple {`sourceID, destinationID, receiptFlag`}. To specify the packet receipt information for all source-destination pairs, one requires $\sum_{i=1}^{S}\sum_{j=1}^{D} a_{ij} = N_{pw}$ of such 3-tuple entries. This requires $N_{pw}\,(2L_I + 1)$ bits, where L_I denotes the number of bits needed to specify a source ID or a destination ID. However, when the same source needs to send its packet to multiple destinations (i.e., multicast) and/or a node is a destination for multiple sources, one can efficiently represent the G-SACK information as an $S \times D$ matrix. The matrix representation would avoid the need for repeating the `sourceID` and the `destinationID`, which consume more bits than the `receiptFlag`. For the matrix implementation, one requires a mapping from the source IDs to the row indices, another mapping from the destination IDs to the column indices, and one bit for each entry of the $S \times D$ matrix. A total of $(S + D)L_I + SD$ bits are required. The savings with the matrix implementation is given by

$$N_{pw}\left(2 - \frac{1}{d^D} - \frac{1}{d^S}\right)L_I + \left(N_{pw} - \frac{N_{pw}^2}{d^S d^D}\right),$$

which increases with d^S and/or d^D.

Update of G-SACKs: The G-SACK matrix of each relay could be different. Ideally, each relay should contain complete "global" information. However, only "local" information is available initially. The "ideal" G-SACK matrix $G = [g_{ij}]$ is an $S \times D$ matrix, where, $\forall i = 1, \ldots, S$, and $\forall j = 1, \ldots, D$, the entry $g_{ij} = 1$ if destination j has already received the packet from source i; otherwise, $g_{ij} = 0$. However, the G-SACK information at a particular relay node might differ from the ideal G-SACK matrix. For example, even after the packet from source i has already been received at destination j (and a corresponding acknowledgment has already been generated), the entry g_{ij} of the (local) G-SACK matrix G_R at some relay node 'R' may still be equal to 0 if 'R' has not yet come in contact with a node having this information.

When all the packets have been received at their intended destinations, and sufficient mixing has been occurred inside the network, the packet receipt information at all the destinations is contained in the G-SACKs. Then, the G-SACK matrix G_R at a relay node 'R' becomes equal to the flow matrix A. For example, for the network in Figure 2 and flow matrix A_U, a G-SACK matrix $G_R = A_U$ at a relay node 'R' indicates that all the packets have been received at their destinations. But, a G-SACK matrix

$$G_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

at relay 'R' indicates that all the packets, except that the packet from source 1 to destination 1, have been received by their destinations. This may happen due to two reasons: (i) either destination 1 has not yet received the packet from source 1, (ii) or the acknowledgment for the said packet has not yet been mixed with the above G-SACK inside the network. Clearly, different relays may contain different G-SACKs.

5. Performance Evaluation Methodology

In this section, we describe the simulation settings and performance metrics that we have used to quantify the gains due to our enhancement features.

5.1. Simulation Setting

We demonstrate the performance benefits due to each enhancement feature of our proposal by constructing appropriate replication/spreading schemes (Sections 6-8), progressively incorporating one enhancement feature at a time, and comparing with the *Basic Scheme* described in the following.

The Basic Scheme: When relay i, which is empty (i.e., relay i has neither a packet nor an acknowledgment), meets with a source, relay i gets a copy of the packet from the source. When relay j, which is empty, meets with relay i, which has a packet, relay j gets the packet. When relay j, which has a packet, meets with a destination (of the packet), the destination gets the packet and the packet in relay j is replaced with an acknowledgment for the currently received packet, henceforth, called an ACK. When relay k, which is empty, meets with relay j, which has an ACK, relay k gets the ACK. When relay k, which has an ACK, meets with a source (which is the intended recipient of that ACK), the source gets the ACK. ■

We developed a customized simulator in MATLAB for DTNs of the type discussed in Section 2. Given any flow matrix, our simulator can simulate the considered schemes. In particular, we compare the schemes by using the topologies shown in Figures 2-6 under a homogeneous as well as a heterogeneous mobility model.

The simulation setting is as follows. We took number of relays $N = 100$ and Galois field size $q = 4$ (when there is coding at relays). We simulated each scheme for $M = 1000$ times with each run for a duration $T = 100$ units of time. In the case of homogeneous mobility, the rate parameter β of the i.i.d. exponentially distributed inter-meeting times is given by $1/\beta = 20$ units of time. As described in Section 2, we appropriately choose the parameters of the heterogeneous mobility so that the network-wide expected meeting rate is equal to β.

5.2. Performance Metrics

We quantify the gains due to each enhancement feature described in Section 4 through the following performance metrics:

Forward delays: We denote the forward delay from source i to destination j for the k-th simulation run by $D_{ij}^f(k)$. It refers to the delay between the sending of the first copy of the packet from source i and the receipt of the first copy at destination j. If destination j does not receive the packet from source i within the simulation time T, then we take $D_{ij}^f(k) = T$. For M simulation runs,

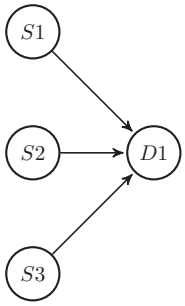

Figure 3: Topology 1

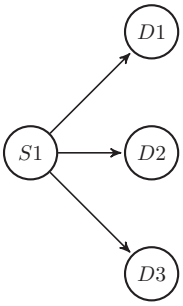

Figure 4: Topology 2

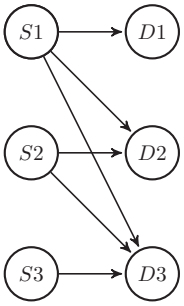

Figure 5: Topology 4

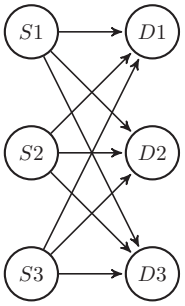

Figure 6: Topology 5

we obtain a vector

$$\mathbf{D}_{ij}^f = \left(D_{ij}^f(1), D_{ij}^f(2), \ldots, D_{ij}^f(M)\right)$$

of M samples for the pairwise forward delay (random variable) D_{ij}^f from source i to destination j.

Next, we discard the samples with value T and obtain a new vector

$$\tilde{\mathbf{D}}_{ij}^f = \left(\tilde{D}_{ij}^f(1), \tilde{D}_{ij}^f(2), \ldots, \tilde{D}_{ij}^f(M_{ij}^f)\right)$$

of M_{ij}^f samples, where M_{ij}^f, $M_{ij}^f \leq M$, denotes the number of simulation runs in which the forward path from source i to destination j is complete within the simulation time T. The expected value $E[D_{ij}^f]$ and the standard deviation $\sigma[D_{ij}^f]$ of the pairwise forward delay (random variable) D_{ij}^f are computed by

$$E[D_{ij}^f] := \frac{1}{M_{ij}^f} \sum_{k=1}^{M_{ij}^f} \tilde{D}_{ij}^f(k), \quad \text{and}$$

$$\sigma[D_{ij}^f] = \sqrt{\frac{1}{M_{ij}^f - 1} \sum_{k=1}^{M_{ij}^f} \left(\tilde{D}_{ij}^f(k) - E[D_{ij}^f(k)]\right)^2}.$$

Clearly, $E[D_{ij}^f]$ and $\sigma[D_{ij}^f]$ represent the *sample mean* and the *sample standard deviation*, respectively, computed using the M_{ij}^f samples corresponding to the simulation runs in which the forward path is complete.

We denote by M_{nw}^f the number of simulation runs in which the forward paths for *all* the (pairwise) flows in the network are complete within the simulation time T. Clearly, $M_{nw}^f \leq M_{ij}^f \leq M$ for all i, j. Restricting to the M_{nw}^f simulation runs, we obtain, for each source-destination pair (i, j), the vector

$$\hat{\mathbf{D}}_{ij}^f = \left(\hat{D}_{ij}^f(1), \hat{D}_{ij}^f(2), \ldots, \hat{D}_{ij}^f(M_{nw}^f)\right)$$

of M_{nw}^f samples. We denote the (network-wide) *average* (resp. *maximum*) forward delay for the k-th simulation run by $D_{avg}^f(k)$ (resp. $D_{max}^f(k)$), which refers to the average (resp. maximum) of the forward delays of all source-destination pairs in the network in the k-th simulation run (given that the forward path for all the flows in the network is complete within the simulation time T), i.e.,

$$D_{avg}^f(k) := \frac{\sum_{i=1}^{S} \sum_{j=1}^{D} a_{ij} \hat{D}_{ij}^f(k)}{N_{pw}}, \quad \text{and}$$

$$D_{max}^f(k) := \max\{\hat{D}_{ij}^f(k) : a_{ij} = 1\},$$

$i = 1, \ldots, S$, $j = 1, \ldots, D$. For the M_{nw}^f simulation runs in which the forward paths for all the (pairwise) flows in the network are complete within the simulation time T, we obtain the vectors

$$\mathbf{D}_{avg}^f = \left(D_{avg}^f(1), D_{avg}^f(2), \ldots, D_{avg}^f(M_{nw}^f)\right), \text{ and}$$

$$\mathbf{D}_{max}^f = \left(D_{max}^f(1), D_{max}^f(2), \ldots, D_{max}^f(M_{nw}^f)\right),$$

for the network-wide average forward delay (random variable) D_{avg}^f and the network-wide maximum forward delay (random variable) D_{max}^f, respectively, each consisting of M_{nw}^f samples. The corresponding expected values and standard deviations are computed as for the pairwise forward delays, but now using the M_{nw}^f samples in $\mathbf{D}_{avg}^f$ and $\mathbf{D}_{max}^f$, respectively.

Return delays: We denote the return delay from destination j to source i for the k-th simulation run by $D_{ij}^r(k)$. It refers to the delay between the receipt of (the first copy of) the packet from source i at destination j and the receipt of the (first copy of) the corresponding acknowledgement at source i. Note that $D_{ij}^r(k)$ is meaningful only if destination j has received the packet from source i. Hence, we restrict to the M_{ij}^f simulation runs

in which the forward path from source i to destination j is complete within the simulation time T, and obtain a vector

$$\mathbf{D}_{ij}^r = \left(D_{ij}^r(1), D_{ij}^r(2), \ldots, D_{ij}^r(M_{ij}^f)\right)$$

of M_{ij}^f samples for the pairwise return delay (random variable) D_{ij}^r. We take the return delay equal to T if the return path for the flow under consideration is not complete within T.

Next, we discard the samples with value T and obtain a new vector

$$\tilde{\mathbf{D}}_{ij}^r = \left(\tilde{D}_{ij}^r(1), \tilde{D}_{ij}^r(2), \ldots, \tilde{D}_{ij}^r(M_{ij}^r)\right)$$

of M_{ij}^r samples, where M_{ij}^r, $M_{ij}^r \leq M_{ij}^f \leq M$, denotes the number of simulation runs in which the return path from destination j to source i is complete (conditioned upon completion of the forward path between the same pair). The expected value $E[D_{ij}^r]$ and the standard deviation $\sigma[D_{ij}^r]$ of the pairwise return delay (random variable) D_{ij}^r are computed as before, but using the M_{ij}^r samples.

We denote by M_{nw}^r the number of simulation runs in which the return paths for *all* the (pairwise) flows in the network are complete within T (conditioned upon the completion of the forward paths for *all* the (pairwise) flows). Clearly, $M_{nw}^r \leq M_{ij}^r \leq M_{ij}^f \leq M$ for all i, j. Restricting to the M_{nw}^r simulation runs, we obtain, for each source-destination pair (i, j), the vector

$$\hat{\mathbf{D}}_{ij}^r = \left(\hat{D}_{ij}^r(1), \hat{D}_{ij}^r(2), \ldots, \hat{D}_{ij}^r(M_{nw}^r)\right)$$

of M_{nw}^r samples. We denote the (network-wide) *average* (resp. *maximum*) return delay for the k-th simulation run by $D_{avg}^r(k)$ (resp. $D_{max}^r(k)$), which refers to the average (resp. maximum) of the return delays of all source-destination pairs in the network in the k-th simulation run (given that the return path for all the flows in the network is complete within the simulation time T), i.e.,

$$D_{avg}^r(k) := \frac{\sum_{i=1}^{S}\sum_{j=1}^{D} a_{ij}\hat{D}_{ij}^r(k)}{N_{pw}}, \quad \text{and}$$

$$D_{max}^r(k) := \max\{\hat{D}_{ij}^r(k) : a_{ij} = 1\},$$

$i = 1, \ldots, S$, $j = 1, \ldots, D$. For the M_{nw}^r simulation runs, similar to forward paths, we obtain the vectors

$$\mathbf{D}_{avg}^r = \left(D_{avg}^r(1), D_{avg}^r(2), \ldots, D_{avg}^r(M_{nw}^r)\right), \text{ and}$$

$$\mathbf{D}_{max}^r = \left(D_{max}^r(1), D_{max}^r(2), \ldots, D_{max}^r(M_{nw}^r)\right),$$

for the network-wide average return delay (random variable) D_{avg}^r and the network-wide maximum return delay (random variable) D_{max}^r, respectively. The corresponding expected values and standard deviations are computed as for the pairwise return delays, but now using the M_{nw}^r samples in $\mathbf{D}_{avg}^r$ and $\mathbf{D}_{max}^r$, respectively.

Round trip delay: It is simply equal to the sum of the forward and return delays. Observing that the the round trip path is complete whenever the return path is complete, we obtain $M_{ij}^{rt} = M_{ij}^r$ samples for the pairwise round trip delay (random variable) D_{ij}^{rt} for flow (i, j). The corresponding expected value and standard deviation are computed as before using the $M_{ij}^{rt} = M_{ij}^r$ samples. Similarly, we obtain $M_{nw}^{rt} = M_{nw}^r$ samples for the network-wide average round trip delay (random variable) D_{avg}^{rt} as well as for the network-wide maximum round trip delay (random variable) D_{max}^{rt}. The corresponding expected values and standard deviations are computed using the $M_{nw}^{rt} = M_{nw}^r$ samples.

Forward success probability: For a source-destination pair (i, j), it refers to the fraction (or percentage) of simulation runs in which the packet reaches from source i to destination j within the simulation time T, and is denoted by P_{ij}^f. By definition, $P_{ij}^f := P(D_{ij}^f \leq T)$, and for M simulation runs, we compute P_{ij}^f by

$$P_{ij}^f = \frac{M_{ij}^f}{M}.$$

The network-wide forward success probability P^f refers to the fraction of simulation runs in which the forward paths for all the flows in the network are successful within simulation time T, computed by

$$P^f = \frac{M_{nw}^f}{M}.$$

Return success probability: For a source-destination pair (i, j), it refers to the fraction of simulation runs in which the acknowledgement reaches from destination j to source i within the simulation time T conditioned upon the completion of the forward path. We denote this probability by P_{ij}^r. By definition,

$$\begin{aligned} P_{ij}^r &:= P(D_{ij}^r \leq (T - D_{ij}^f) | D_{ij}^f \leq T) \\ &= P(D_{ij}^f + D_{ij}^r \leq T) | P(D_{ij}^f \leq T), \end{aligned}$$

and we compute P_{ij}^r by

$$P_{ij}^r = \frac{M_{ij}^r}{M_{ij}^f}.$$

The network-wide return success probability P^r refers to the fraction of simulation runs in which the return paths for all flows under consideration are complete within simulation time T conditioned upon completion of the forward path for all flows, and we compute P^r by

$$P^r = \frac{M_{nw}^r}{M_{nw}^f}.$$

Round-trip success probability: It refers to the fraction of simulation runs in which the round trip path is complete within the simulation time T. It is easy to see that the round-trip success probability is the product of the forward and return success probabilities. For a source-destination pair (i, j), by definition, the round-trip success probability P_{ij}^{rt} is given by

$$\begin{aligned} P_{ij}^{rt} &:= P(D_{ij}^{rt} \leq T) = P(D_{ij}^f + D_{ij}^r \leq T) \\ &= P_{ij}^f P_{ij}^r = \frac{M_{ij}^{rt}}{M}. \end{aligned}$$

Similarly, the network-wide round trip success probability P^{rt} is computed by

$$P^{rt} = \frac{M_{nw}^{rt}}{M}.$$

6. Improving the Return Path

In this section, we show the benefits of 'selective acknowledgments' and 'mixing of acknowledgment information inside the network' which result in improvement in the return path. It is important to remember that, in this section, there is no coding of packets in the forward path, since we are interested in studying the benefits due to "smart" acknowledgment mechanisms alone.

Convention: We adopt the following convention. (i) When each unit of acknowledgment generated by the destinations contains packet receipt information intended for a single source (irrespective of the source degree of the destination that generates it) and there is no mixing of acknowledgment information inside the network, we call it the (plain) ACK Scheme. (ii) When the acknowledgments generated by destinations with source degree greater than one may contain packet receipt information intended for multiple sources and there is no mixing of acknowledgment information inside the network, we add the prefix 'S' with ACK and call it the "Selective" ACK (SACK) Scheme. (iii) When each unit of acknowledgment generated by the destinations contains packet receipt information intended for a single source (irrespective of the source degree of the destination that generates it) and there is mixing of acknowledgment information inside the network, we add the prefix 'G' and call it the "Global" ACK (G-ACK) Scheme. (iv) When the acknowledgments generated by destinations with source degree greater than one may contain packet receipt information intended for multiple sources and there is mixing of acknowledgment information inside the network, we add the prefixes 'G' and 'S' and call it the G-SACK Scheme.

6.1. Benefits of Selective ACK (SACK) over ACK

The SACK Scheme: This scheme differs from the Basic Scheme in only one aspect, namely, a destination with source degree greater than one generates a SACK (instead of an ACK) indicating "the set of packets" it has successfully received so far. ■

In order to observe the benefits of SACKs over ACKs, we consider the example network shown in Figure 3 (we call it Topology 1). In this example network, there are three sources and one destination (common to all sources). Each source unicasts its packet to the destination. For this topology, the average source-degree and destination-degree is equal to 3 and 1, respectively.

In Figure 7, we compare the Cumulative Distribution Functions (CDFs) of the pairwise return delays obtained with the Basic Scheme and the SACK Scheme. In Figure 8, we compare the CDFs for the network-wide maximum return delays. The CDFs corresponding to the homogeneous (resp. heterogeneous) mobility model are denoted by *'exponential'* (resp. *'power law with cutoff'*) to emphasize the distribution of the aggregate inter-meeting time for the two mobility models (refer to our mobility models in Section 2). We follow this convention in all the figures in the remainder of the paper. Referring to Figures 7 and 8, we make the following observations.

The delay CDFs with SACKs stay above the delay CDFs with ACKs (for homogeneous as well as heterogeneous mobility), which implies that

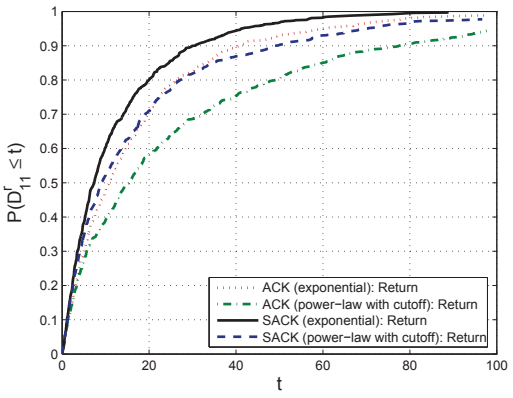

Figure 7: Comparison of pairwise return delay CDF of ACK versus SACK: Topology 1.

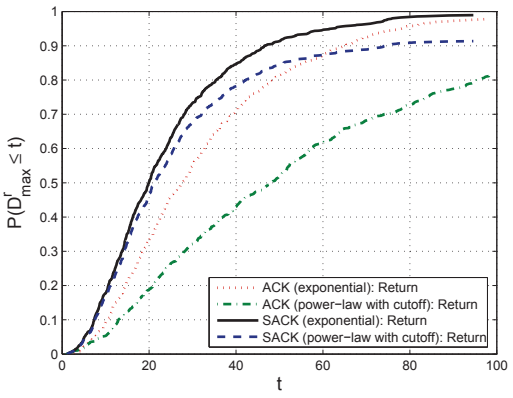

Figure 8: Comparison of network-wide maximum return delay CDF of ACK versus SACK: Topology 1.

O1: *The return delays with SACKs are stochastically smaller [51] than that with ACKs.*

Observation O1 implies that the mean and variance (and all higher moments) of the return delay improve (i.e., decrease) with the SACK Scheme. For example, for homogeneous (resp. heterogeneous) mobility, we observe approximately 25.5%, 28.8% and 24.5% (resp. 28.25%, 32.3%, and 26%) improvement (due to SACKs) in the mean of the pairwise, network-wide average and maximum return delays, respectively. The improvement in standard deviation is approximately 13%-28.3%.

Also, the success probability within any time budget t (on the *x-axis*) is higher with SACK than that with ACK. For example, with homogeneous mobility, at time $t = 20$, P_{11}^r is 0.8 with SACK as compared to 0.7 with ACK (see Figure 7). The improvement in network-wide return success probability (for a time budget $t = 100$) with homogeneous (resp. heterogeneous) mobility is approximately 1% (resp. 12.45%).

The delay CDF with ACKs/SACKs for homogeneous mobility stays above the corresponding delay CDF for heterogeneous mobility, which implies that

O2: *Heterogeneous mobility has an adverse impact on the return delay.*

However, the improvement in return delay due to SACKs is larger for heterogeneous mobility, especially for network-wide metrics. This is evident from the gap between the CDFs – the more the gap the more the improvement (see Figure 8).[4] The reason is that

O3: *The adverse impact of heterogeneous mobility is much more with the ACK Scheme than with the SACK Scheme (see Figure 8).*

6.2. Benefits of Global-ACK (G-ACK) over ACK

The G-ACK Scheme: The G-ACK Scheme differs from the Basic Scheme by the following additional features. When relay i, which has an ACK of one destination meets with another relay j, $j \neq i$, which carries an ACK of another destination, both relays combine their ACK information to form a G-ACK. When a relay j carrying a G-ACK meets with another relay k which carries a(n) G-ACK/ACK containing acknowledgment information not already contained in the G-ACK of relay j, the acknowledgment information are combined together to form a new G-ACK which is then carried by both relay j and k. When a relay carrying a(n) G-ACK/ACK meets with a destination, the destination updates the G-ACK/ACK by including its latest packet receipt information. ■

In order to observe the benefits of G-ACK over ACK, we consider the example network shown in Figure 4 (we call it Topology 2). In this network, there is only one source node which is multicasting its packet to three destination nodes. Each destination generates a (plain) ACK. The average source-degree and destination-degree is equal to 1 and 3, respectively.

In Figure 9, we compare the CDFs of the pairwise return delays obtained with the (plain) ACK Scheme and the G-ACK Scheme. In Figure, 10, we

[4]The gap between two CDFs can be precisely measured by *Kolmogorov-Smirnov distance*, *Kullback-Leibler distance*, or other divergence measures [52].

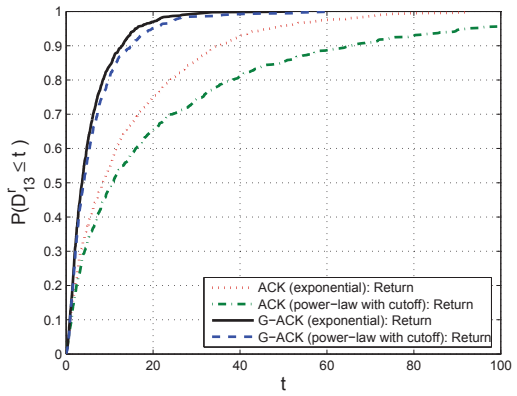

Figure 9: Comparison of pairwise return and round trip delay CDF with the (plain) ACK Scheme and the G-ACK Scheme: Topology 2.

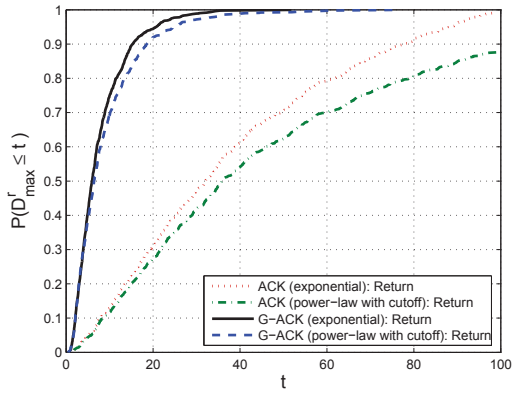

Figure 10: Comparison of network-wide maximum round trip delay CDF of the (plain) ACK Scheme and the G-ACK Scheme: Topology 2.

compare the CDFs of the network-wide maximum return delays. Referring to Figures 9 and 10, we make the following observations.

O4: *The return delay with G-ACK is stochastically smaller than the return delay with ACK.*

The gap between the CDFs is more in this case as compared to that for SACK over ACK, implying larger improvements in mean, variance and success probabilities which is mainly due to mixing of ACK information inside the network. For example, for homogeneous (resp. heterogeneous) mobility, we observe approximately 60.6%, 61.4% and 71.75% (resp. 65.6%, 64.5%, and 75.4%) improvement (due to G-ACKs) in the mean of the pairwise, network-wide average and maximum return delays, respectively. The improvement in standard deviation is 53.2%-68.4%. The improvement in network-wide return success probability (for a time budget $t = 100$) with homogeneous (resp. heterogeneous) mobility is 0.91% (resp. 14.15%).

The gains due to G-ACK in the pairwise return delay is small for small time budgets. As time progresses, the relays carrying different ACKs from different destinations start mixing their information. This results in the increase in the number of relays carrying ACK information from multiple destinations, and the benefits start showing up. However, the improvement in the network-wide maximum return delay starts relatively quickly. In fact, without mixing inside the network, the source has to wait for the "last" ACK for a significantly long time. But, with mixing inside the network, there are many relays having ACK information from multiple destinations and the source does not have to wait so long for the "last" ACK. In summary,

O5: *The improvements due to G-ACK is more for network-wide maximum delays than for pairwise delay, and are achieved sooner, i.e., with a smaller time budget t.*

6.3. Benefits of G-SACK over SACK

In this section, we show the benefits of G-SACK over SACK. The idea of G-SACKs is that the destination(s) generate SACK(s) and the SACK(s) are updated inside the network to form G-SACKs.

The G-SACK Scheme: Destinations generate SACKs upon receiving packets, as in the SACK Scheme. But, the G-SACK Scheme differs from the SACK Scheme as follows. When two relays, each carrying a SACK, meet, the SACKs at the relays are combined to form a G-SACK which contains the union of acknowledgment information contained in the two SACKs and both relays carry that G-SACK. When a relay carrying a G-SACK meets with another relay with a G-SACK/SACK, both relays combine their information and carry the latest information. When a relay carrying a G-SACK meets with a destination, the destination updates the G-SACK by including its latest packet receipt information. ■

6.3.1. Benefits of G-SACK over SACK with a Single Destination

We observed the benefits of G-SACK over SACK even with a single destination. We do not report the results here (since there are no new insights to be gained in this case apart from observations similar

to that in Section 6.1 and 6.2). However, we explain by an example why this benefit is gained.

Example 6.1. Consider the following sequence of meetings in order to observe the benefits of G-SACK over SACK. Let a relay $R1$ carrying the packet of Source 1 meets with the destination and brings ACK 1. Let another relay $R2$ carrying the packet of Source 2 meets with the destination and brings a SACK for Source 1 and Source 2. Let both relays meet with each other before coming in contact with any source(s). Let $R1$ meets with Source 2 before $R2$ meets with Source 2. If there is no mixing of SACKs inside the network, then the meeting of $R1$ with Source 2 would not be useful as it has ACK information only for Source 1. But if there is mixing of SACK information, then $R1$ would get the information about the receipt of the packet of Source 2 when it meets with $R2$ and $R1$'s meeting with Source 2 would be beneficial. ■

6.3.2. Benefits of G-SACK over SACK with Multiple Destinations

The real benefits of G-SACK over SACK are observed by considering a network with multiple destinations containing many flows (i.e., under heavy loads). Recalling that a G-SACK can acknowledge *multiple sources* about the receipt of packets at *multiple destinations*, we consider the network shown in Figure 6 to show the benefits of G-SACK over SACK with multiple destinations. In this Topology, there are three sources and three destinations. Each source node multicasts its packet to all three destination nodes. The average source/destination-degree is equal to 3.

In Figure 11, we compare the CDFs of the pairwise return delays with G-SACK and SACK. In Figure 12, we compare the network-wide maximum return delay CDFs. Referring to Figures 11 and 12, we make the following observations.

O6: *The delays with G-SACK are stochastically smaller than with SACK (for both homogeneous and heterogeneous mobility).*

The improvement in delay performance due to mixing of acknowledgment information inside the network increases as the number of multicast sessions increases (compare Figures 10 and 12 which pertain to one and three multicast session(s), respectively). This increase in improvement is due to the fact that a multicast session must continue until acknowledgments from all the destinations are received and G-SACKs help in this regard.

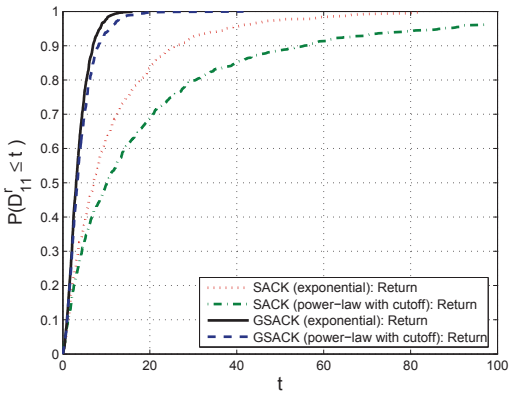

Figure 11: Comparison of pairwise return delay CDF of SACK versus G-SACK (multiple destinations): Topology 5.

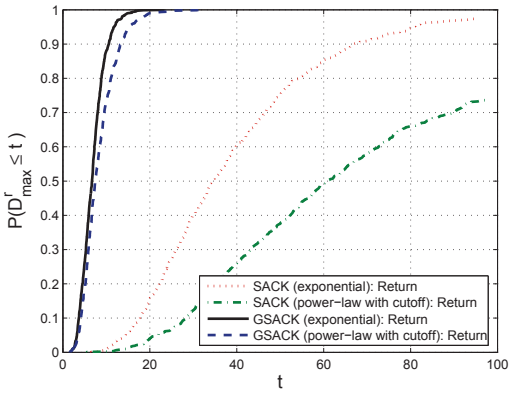

Figure 12: Comparison of network-wide maximum return delay CDF of SACK versus G-SACK (multiple destinations): Topology 5.

In general, the delay performance degrades with load (i.e., with more flows). But, the performance degradation with load is much less when there is mixing of acknowledgment information inside the network. As a result,

O7: *The improvement due to mixing of acknowledgment information inside the network increases with load. In particular, it increases with the number of multicast flows.*

This is evident when we compare Figures 10 and 12 for heterogeneous mobility (recall that the more the gap between the CDFs the more the improvement). For example, for homogeneous (resp. heterogeneous) mobility, we observe approximately 68%, 70% and 81.7% (resp. 75%, 74.6%, and 83.8%) improvement (due to G-SACKs) in the mean of the

pairwise, network-wide average and maximum return delays, respectively. The improvement in standard deviation is 67.9%-84.8%. The improvement in network-wide return success probability (for a time budget $t = 100$) with homogeneous (resp. heterogeneous) mobility is 2.59% (resp. 35.6%).

In fact, Observations O3, O5 and O7, combined together, lead to a generalized observation that the improvement in return delay due to 'selective acknowledgment' and/or 'mixing of acknowledgment information inside the network' is more when the conditions are more demanding or adverse such as with heterogeneous mobility, higher load, requirement of network-wide flow completion or multicasting (which requires receipt of acknowledgments of "all" constituent flows). The reason is that

O8: *The impact of adverse conditions is much less with the enhancements (SACK and/or mixing) in place than when they are not present.*

7. Improving the Forward Path

In this section, we quantify the benefits of inter-session network coding with a Coding Scheme described in the following.

The Coding Scheme: Packet payload parts of different sources are combined at relays by forming random linear combinations (RLCs) over the Galois field $\mathbb{F}_q$ [53, 54]. When relay j, which has a pure or a coded packet, meets with a source or with another relay that has a pure or a coded packet, relay j replaces its content with a new RLC. When an empty relay j meets with a source or with another relay that has a pure or a coded packet, relay j copies the pure or coded packet. ■

To quantify the benefits of coding, only the forward path is relevant. We quantify the gains due to coding by comparing the forward success probabilities and the CDFs of the forward delay (random variables) obtained with the Coding Scheme and the Basic Scheme (in which the forward path does not involve coding (see Section 5)). Recall that we have taken Galois field size $q = 4$. In fact, we observed the forward delays to improve when the Galois field size was increased from $q = 2$ to $q = 4$. However, with the small number of flows considered in this paper, the forward delays were found to be not improving any more with further increase in q.

In Figures 13 and 14, we compare the CDFs of the pairwise forward delay and the network-wide maximum forward delay, respectively, obtained with the

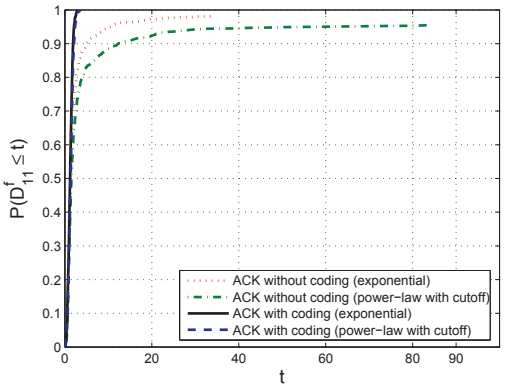

Figure 13: Comparison of pairwise forward delay CDFs with and without coding: Topology 3.

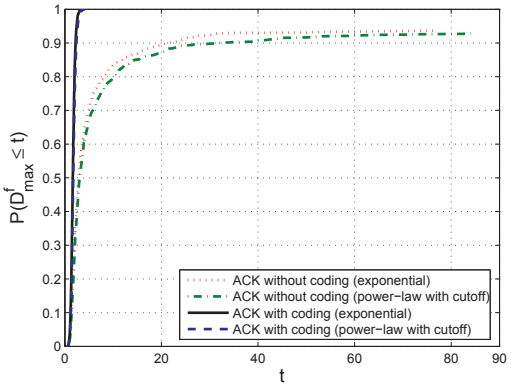

Figure 14: Comparison of network-wide maximum forward delay CDFs with and without coding: Topology 3.

Coding Scheme and the Basic Scheme (i.e., without coding) for the topology shown in Figure 2 consisting of 3 source-destination pairs. In Figure 2, each source unicasts its packet to one of the destination nodes and the average source/destination-degree is equal to 1. In Figures 15 and 16, we provide results for a similar unicast topology with 10 source-destination pairs. Referring to Figures 13-16, we make the following observations.

For the case with 3 unicast flows, for homogeneous (resp. heterogeneous) mobility, the pairwise forward success probability, P^f_{11}, without coding is observed to be 98.1% (resp. 95.5%) and the network-wide forward success probability P^f = 93.7% (resp. 92.8%). We emphasize that P^f_{11} and P^f are less than 100%, even with a load of only 3 unicast flows, primarily due to Type-II non-

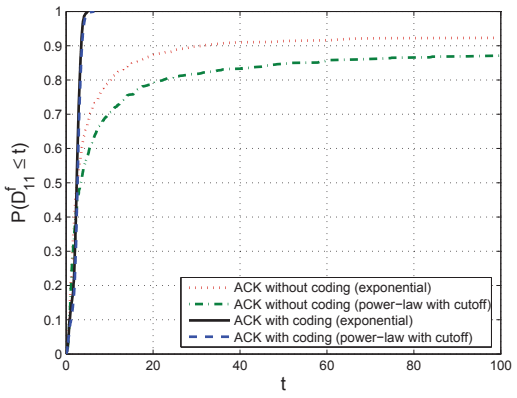

Figure 15: Comparison of pairwise forward delay CDFs with and without coding.

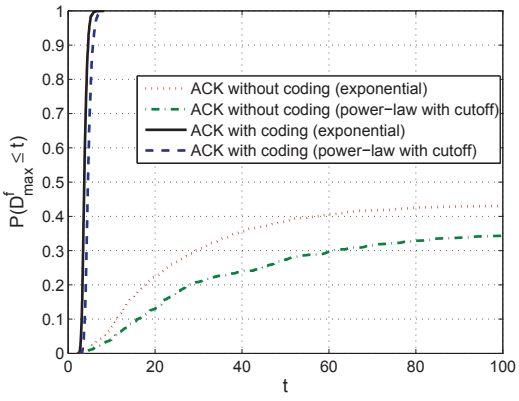

Figure 16: Comparison of network-wide maximum forward delay CDFs with and without coding.

reachability (recall the Type-I and Type-II non-reachability explained in Section 3). We have observed from simulations that even a five-fold increase of the simulation time from $T = 100$ to $T = 500$ does not improve the forward success probabilities any further. In fact, it can be observed in Figure 13 that, for homogeneous mobility without coding, the flow $(1, 1)$ is complete within 40 units of time, if it does. Similarly, it can be observed in Figure 14 that, for homogeneous mobility without coding, all flows are complete within 80 units of time, if they do. However, with coding, the forward success probabilities become 100%.

For a higher load of 10 unicast flows, for the case without coding, the simulation time $T = 100$ is short enough to cause Type-I non-reachability as well; observe that the CDFs for the 'without coding' case in Figures 15 and 16 go until a delay value $t = 100$. However, with coding, the forward success probabilities are 100%. Moreover, coding provides 100% successful completion within a much smaller time budget of ≈ 10 time units. In summary,

O9: *Coding reduces Type-I non-reachability. Moreover, coding rules out Type-II non-reachability of data packets; sooner or later each source meets with some relay and its packet enters into the network (although in coded form) and finally gets delivered.*

A 132% improvement (2.32 times increase) (resp. 182.5% improvement (2.825 times increase)) in network-wide success probability for homogeneous (resp. heterogeneous) mobility is indeed remarkable. The improvement in the mean of the forward delay for homogeneous (resp.heterogeneous) mobility is 56.76%-84.77% (resp. 59.8%-85.4%). The 89.96%-96.7% (resp. 90.94%-96.64%) improvement in the standard deviation of forward delays for homogeneous (resp.heterogeneous) mobility is also remarkable, which implies that

O10: *Coding makes the packet transfers smoother.*

We have observed that the forward delay CDFs with and without coding intersect each other and the forward delay CDF without coding stays below the forward delay CDF with coding for a negligible initial part (see Figure 15).

This can be explained as follows. Coding involves a small delay for the packets from various sources to get mixed. Also the destination needs, on the average, more than one coded packets to extract its required packet whereas exactly one (uncoded or pure) packet is required without coding. The benefits of coding come into play after a small delay, which we call the *mixing delay*, after which there is a proper mixture of coded and uncoded packets in the network. The mixing delay can be measured by the delay value on the t-axis (*x-axis*) when the two CDFs intersect each other. Without coding, some packet might reach its destination very quickly due to randomness and this accounts for the CDF without coding staying above the CDF with coding before the mixing delay.

For homogeneous mobility, the ratio of maximum to average expected forward delay decreases from 2.07 (without coding) to 1.37 (with coding) in case of 3 unicast flows. Similarly, it decreases from 4.36 (without coding) to 1.55 (with coding) in case of 10 unicast flows. Such decrease is even larger with heterogeneous mobility. This implies that

O11: *Coding improves the fairness among the packet transfers, and the improvement in fairness increases with increase in the load.*

8. Benefits of our overall Proposed Scheme over the Basic Scheme

In this section, we show the benefits of our overall proposal over the Basic Scheme.

The Proposed Scheme: Our Proposed Scheme consists of coding to improve the forward path, and G-SACKs to improve the return path. ■

We consider three different topologies in order to show the benefits of our Proposed Scheme against the Basic Scheme (which uses plain ACKs without coding of packets). The topologies considered are Topology 3, Topology 4 and Topology 5 shown in Figures 2, 5 and 6, respectively. Note that the average source and destination degrees in Topology 3, 4, and 5 are 1, 2, and 3, respectively.

In Figures 17, 18 and 19, we compare the CDFs of the network-wide maximum round trip delays obtained with the Basic Scheme and the Proposed Scheme under Topology 3, 4, and 5, respectively. Referring to Figures 17-19, we make the following observations.

The round trip delays with our Proposed Scheme are stochastically significantly smaller than that with the Basic Scheme (Figures 17-19). The reduction in the mean and standard deviation of the round trip delay with our Proposed Scheme as compared to the Basic Scheme is indeed remarkable (see Tables 7, 8 and 9 provided as online supplementary material). It is also worth emphasizing that

O12: *The round-trip success probability with our Proposed Scheme for all three considered topologies is equal to 1.*

The performance of the Basic Scheme degrades with the increase in the source and destination degrees. However,

O13: *With our Proposed Scheme, the return delays, in fact, improve with the increase in the source degree.*

This reduction, which appears surprising at first, can be explained as follows. On receiving (resp. extracting) the first copy of each packet in the absence of coding (resp. with network coding), a destination replaces the packet (resp. RLC) with a corresponding ACK/SACK, henceforth called a "seed", which triggers its own "chain reaction" of creating

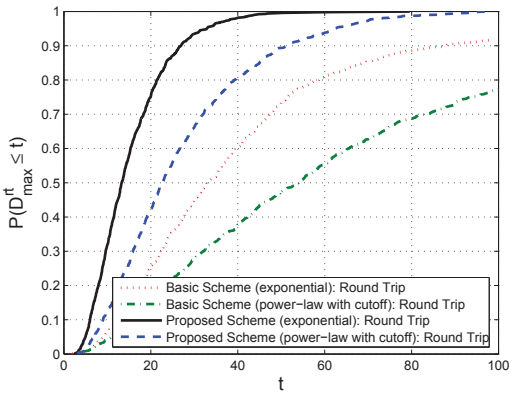

Figure 17: Comparison of network-wide maximum round trip delay CDF with the Basic Scheme versus the Proposed Scheme: Topology 3.

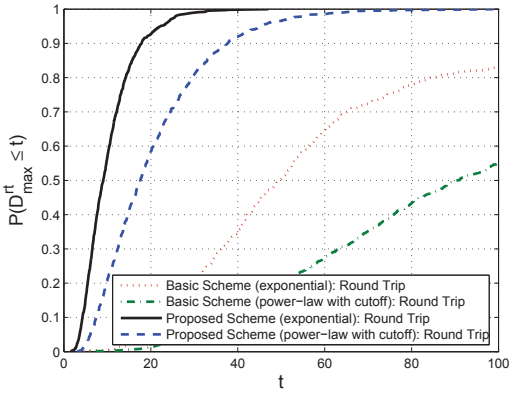

Figure 18: Comparison of network-wide maximum round trip delay CDF with the Basic Scheme versus the Proposed Scheme: Topology 4.

more acknowledgments. If the destinations generate SACKs instead of ACKs, then the number of "seeds" generated by a destination for a given packet (i.e., the number of SACKs containing the receipt information of that packet) increases with the increase in its source degree; each time a new packet is received/extracted, a SACK containing information about all the packets received so far is generated. There are more "seeds" for the first received/extracted packet than for the second, and so on. Moreover, with the (plain) ACK Scheme the chain reactions corresponding to different ACKs compete to get access to the relay buffers, but with G-SACKs there is no competition. So, the return paths with G-SACKs become faster with the increase in the source degree due to the increase in

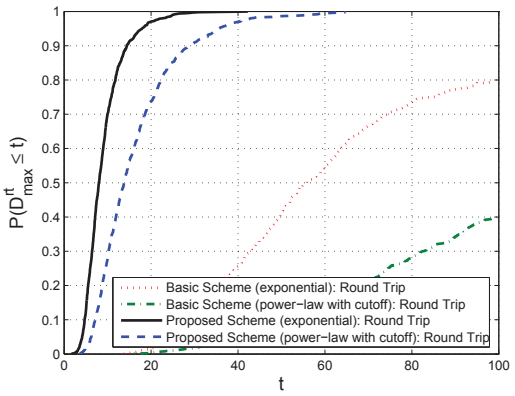

Figure 19: Comparison of network-wide maximum round trip delay CDF with the Basic Scheme versus the Proposed Scheme: Topology 5.

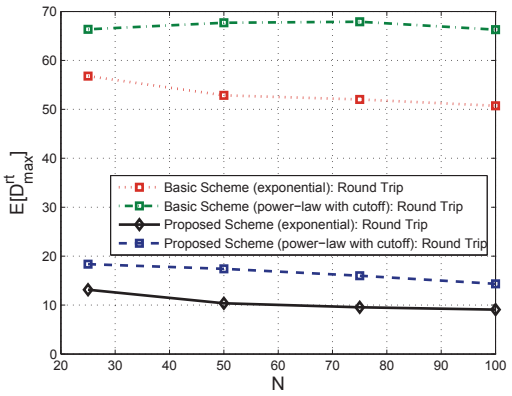

Figure 20: Comparing network-wide mean forward and mean round trip delay with the Basic Scheme and the Proposed Scheme: Topology 5.

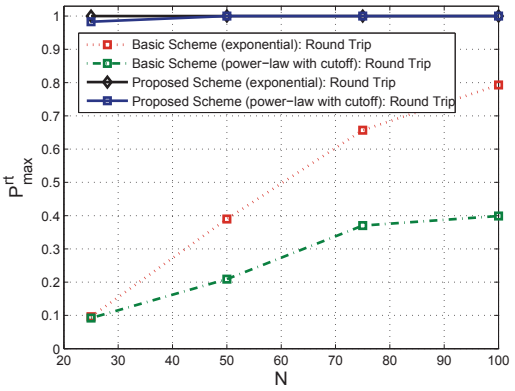

Figure 21: Comparing network-wide forward and round trip success probability with the Basic Scheme and the Proposed Scheme: Topology 5.

the number of "seeds" (SACKs) for the same acknowledgment information.

8.1. Effect of Number of Nodes

Next, we obtain and compare additional simulation results with $N = 25, 50, 75$. But, as suggested in [47], we keep the product $N\beta$ as constant, which ensures that the total rate at which meetings occur in the network remain constant so that one can compare the cases with different number of nodes.

In Figures 20 and 21, we compare the mean of the network-wide maximum round trip delays and the corresponding success probabilities, respectively, obtained with the Basic Scheme and the Proposed Scheme by varying the number of relays N. Referring to Figures 20 and 21, we make the following observations.

O14: *The decrease in the mean of the network-wide maximum round trip delays with our Proposed Scheme as compared to the Basic Scheme are indeed remarkable for all values of* $N = 25, 50, 75, 100$.

O15: *The network-wide round trip success probabilities with our Proposed Scheme are equal to 1 for all values of* $N = 25, 50, 75, 100$ *(except for* $N = 25$ *with heterogeneous mobility when it is* 0.98*). This shows the effectiveness of our overall scheme in ensuring reliability.*

Remark: It is worth mentioning that, with homogeneous mobility, our Proposed Scheme achieves a round-trip success probability equal to 1 even with 10 different source nodes each multicasting its packet to 10 destination nodes (results not reported here) and even with number of relays equal to 50.

9. Conclusion

In this paper, we proposed and studied through extensive simulations a set of enhancement features to improve reliable transport in terrestrial DTNs consisting of both unicast and multicast flows. We observed that random linear network coding makes the forward component more reliable (i.e., increases the forward success probability) and the G-SACK makes the return path more reliable.

We have quantified the benefits due to each enhancement feature under appropriately chosen "minimal" topologies. Our extensive simulations

(not reported here) indicate that the benefits due to each enhancement feature improves even further with bigger networks with larger source-destination degrees. We also quantified the overall gain achieved by all of the features together.

Our Proposed Scheme, which combines random linear network coding of packets and G-SACKs, provides an improvement, with homogeneous (resp. heterogeneous) mobility, up to 26.1%, 82.1% and 75.4% (resp. 150.6%, 78.4% and 56.2%) in the network-wide round trip success probability, and the mean and standard deviation of the network-wide maximum round trip delay, respectively, as compared to the Basic Scheme. Our main insights is that this substantial improvement comes from the *mixing of information inside the network* which essentially solves the two primary limitations of DTNs, namely, *randomness* and *finite capacity*.

We have observed that random linear network coding makes the forward path smoother, i.e., it decreases delay variance, and improves the fairness among flows. A more detailed evaluation with multiple packet buffers and multiple packet transfers between source-destination pairs is part of our ongoing work.

References

[1] A. Ali, T. Chahed, E. Altman, M. Panda, L. Sassatelli, A new proposal for reliable unicast and multicast transport in delay tolerant networks, in: Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on, IEEE, pp. 1129–1134.

[2] http://www.dtnrg.org, Internet Research Task Force Delay-Tolerant Networking Research Group, 2012.

[3] C. Caini, H. Cruickshank, S. Farrell, M. Marchese, Delay-and disruption-tolerant networking (DTN): An alternative solution for future satellite networking applications, Proceedings of the IEEE (2011) 1–18.

[4] P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, C. Cervelló-Pastor, From delay-tolerant networks to vehicular delay-tolerant networks, Communications Surveys & Tutorials, IEEE (2011) 1–17.

[5] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, C. Diot, Pocket switched networks and human mobility in conference environments, in: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, pp. 244–251.

[6] T. Jonson, J. Pezeshki, V. Chao, K. Smith, J. Fazio, Application of delay tolerant networking (DTN) in airborne networks, in: Military Communications Conference, 2008. MILCOM 2008. IEEE, IEEE, pp. 1–7.

[7] N. Kayastha, D. Niyato, P. Wang, E. Hossain, Applications, architectures, and protocol design issues for mobile social networks: A survey, Proceedings of the IEEE 99 (2011) 2130–2158.

[8] T. Small, Z. Haas, The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way), in: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, ACM, pp. 233–244.

[9] A. Pentland, R. Fletcher, A. Hasson, Daknet: rethinking connectivity in developing nations, Computer 37 (2004) 78–83.

[10] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, Delay-tolerant networking: an approach to interplanetary internet, Communications Magazine, IEEE 41 (2003) 128 – 136.

[11] G. Holland, N. Vaidya, Analysis of TCP Performance Over Mobile Ad Hoc Networks, ACM Wireless Networks 8 (2002) 275–288.

[12] M. Ramadas, S. Burleigh, S. Farrell, Licklider Transmission Protocol - Motivation, in: RFC-5325.

[13] H. Balakrishnan, S. Seshan, E. Amir, R. Katz, Improving TCP/IP performance over wireless networks, in: Proceedings of the 1st annual international conference on Mobile computing and networking, ACM, pp. 2–11.

[14] H. Balakrishnan, et al., Challenges to reliable data transport over heterogeneous wireless networks, University of California, Berkeley, 1998.

[15] T. Mahmoodi, V. Friderikos, O. Holland, A. Hamid Aghvami, Cross-layer design to improve wireless tcp performance with link-layer adaptation, in: Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th, IEEE, pp. 1504–1508.

[16] S. Biswas, R. Morris, ExOR: Opportunistic Multi-Hop Routing for Wireless Networks, in: Sigcomm.

[17] T. Spyropoulos, K. Psounis, C. Raghavendra, Efficient Routing in Intermittently Connected Mobile Networks: The Multi-copy Case, IEEE/ACM Transaction on Networking 16 (2008) 63–76.

[18] A. Lindgren, A. Doria, O. Schelen, Probabilistic Routing in Intermittently Connected Networks, in: ACM Mobicom.

[19] M. Grossglauser, D. Tse, Mobility increases the capacity of ad hoc wireless networks, Networking, IEEE/ACM Transactions on 10 (2002) 477–486.

[20] A. Vahdat, D. Becker, Epidemic Routing for Partially-Connected Ad Hoc Networks, in: Duke University, Tech. Rep. CS-200006.

[21] K. Scott, S. Burleigh, Bundle Protocol Specification, in: RFC-5050.

[22] L. Wood, J. McKim, W. Eddy, W. Ivancic, C. Jackson, Saratoga: A Scalable File Transfer Protocol.

[23] M. Ramadas, S. Burleigh, S. Farrell, Licklider Transmission Protocol Specification, in: RFC-5326.

[24] CCSDS, CCSDS File Delivery Protocol (CFDP), in: CCSDS 727.0-B-4, Blue Book.

[25] G. Papastergiou, I. Psaras, V. Tsaoussidis, Deep-Space Transport Protocol: A Novel Transport Scheme for Space DTNs, Computer Communications, Special Issue of Computer Communicationson Delay and Disruption Tolerant Networking 32 (2009) 1757–1767.

[26] O. B. Akan, J. Fang, I. F. Akyildiz, TP-Planet: A Reliable Transport Protocol for InterPlanetary Internet, IEEE Journal on Selected Areas in Communications(JSAC) 22 (2004) 348–361.

[27] C. C. for Space Data Systems, Space Communications Protocol Standards (SCPS) - Transport Protocol (SCPS-TP), in: CCSDS 714.0-B-2, Blue Book.

[28] K. A. Harras, K. C. Almeroth, Transport Layer Issues

in Delay Tolerant Mobile Networks, in: IFIP NETWORKING.
[29] K. Fall, S. McCanne, You don't know jack about network performance, Queue 3 (2005) 54–59.
[30] R. Ahlswede, N. Cai, S.-Y. Li, R. Yeung, Network information flow, Information Theory, IEEE Transactions on 46 (2000) 1204 –1216.
[31] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, B. Leong, A random linear network coding approach to multicast, Information Theory, IEEE Transactions on 52 (2006) 4413 –4430.
[32] S.-Y. Li, R. Yeung, N. Cai, Linear network coding, Information Theory, IEEE Transactions on 49 (2003) 371 –381.
[33] R. Koetter, M. Medard, An algebraic approach to network coding, Networking, IEEE/ACM Transactions on 11 (2003) 782 – 795.
[34] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, Xors in the air: practical wireless network coding, in: ACM SIGCOMM Computer Communication Review, volume 36, ACM, pp. 243–254.
[35] Y. Lin, B. Li, B. Liang, Efficient network coded data transmissions in disruption tolerant networks, pp. 1508–1516.
[36] S. Sengupta, S. Rayanchu, S. Banerjee, An analysis of wireless network coding for unicast sessions: The case for coding-aware routing, in: INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, IEEE, pp. 1028–1036.
[37] J. Widmer, J.-Y. Le Boudec, Network coding for efficient communication in extreme networks, in: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, WDTN '05, ACM, New York, NY, USA, 2005, pp. 284–291.
[38] X. Zhang, G. Neglia, J. Kurose, D. Towsley, On the benefits of random linear coding for unicast applications in disruption tolerant networks, in: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on, IEEE, pp. 1–7.
[39] Y. Lin, B. Liang, B. Li, Performance modeling of network coding in epidemic routing, in: Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking, ACM, pp. 67–74.
[40] E. Altman, F. D. Pellegrini, L. Sassatelli, Dynamic Control of Coding in Delay Tolerant Networks, in: Infocom.
[41] A. Ali, E. Altman, T. Chahed, M. Panda, L. Sassatelli, A new reliable transport scheme in delay tolerant networks based on acknowledgments and random linear coding, in: ITC 24.
[42] M. Liu, Y. Yang, Z. Qin, A survey of routing protocols and simulations in delay-tolerant networks, in: Y. Cheng, D. Eun, Z. Qin, M. Song, K. Xing (Eds.), Wireless Algorithms, Systems, and Applications, volume 6843 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2011, pp. 243–253.
[43] T. Karagiannis, J.-Y. Le Boudec, M. Vojnović, Power law and exponential decay of inter contact times between mobile devices, in: Proceedings of the 13th annual ACM international conference on Mobile computing and networking, ACM, pp. 183–194.
[44] A. Passarella, M. Conti, Analysis of individual pair and aggregate inter-contact times in heterogeneous opportunistic networks, Transactions on Mobile Computing, IEEE (accepted for publication).
[45] R. Groenvelt, P. Nain, G. Koole, The Message Delay in Mobile Ad Hoc Networks, Performance Evaluation 62 (2005) 210–228.
[46] M. Ibrahim, A. A. Hanbali, P. Nain, Delay and Resource Analysis in MANETs in Presence of Throwboxes, Performance Evaluation 24 (2007) 933–945.
[47] X. Zhang, G. Neglia, J. Kurose, D. Towsley, Performance Modeling of Epidemic Routing, Computer Networks 51 (2007) 2867–2891.
[48] M. E. Newman, Power laws, pareto distributions and zipf's law, Contemporary physics 46 (2005) 323–351.
[49] A. Krifa, C. Barakat, T. Spyropoulos, Mobitrade: trading content in disruption tolerant networks, in: Proceedings of the 6th ACM workshop on Challenged networks, ACM, pp. 31–36.
[50] A. Krifa, C. Barakat, T. Spyropoulos, et al., Message drop and scheduling in DTNs: Theory and practice (2010).
[51] M. Shaked, J. Shanthikumar, Stochastic orders and their applications, Probability and mathematical statistics, Academic Press, 1994.
[52] O. Thas, Comparing distributions, Springer, 2010.
[53] P. Morandi, Field and Galois theory, volume 167, Springer Verlag, 1996.
[54] J. Milne, Fields and Galois Theory, Courses Notes, Version 4 (2003).

Table

Table 1: SACK over ACK

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
ACK (Hom)	0.982	2.3758	4.6167	0.9898	16.4023	17.6114	0.972	18.6968	18.1203
SACK (Hom)	0.985	2.2712	3.9634	0.9969	12.2160	13.8255	0.982	14.4364	14.8146
Improvement(%)	—	—	—	0.72	25.52	21.50	1.03	22.79	18.24
ACK (Het)	0.974	3.4828	7.2682	0.9435	22.4059	23.7220	0.9190	25.7949	24.3058
SACK (Het)	0.968	3.3283	6.3234	0.9773	16.0768	18.6827	0.946	19.2518	19.7812
Improvement(%)	—	—	—	3.5760	28.2475	21.2431	2.9380	25.3659	18.6153
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
ACK (Hom)	0.945	2.5167	3.0573	0.9788	16.6185	9.6863	0.925	19.0592	10.1628
SACK (Hom)	0.945	2.2197	2.1982	0.9894	11.8276	6.9466	0.935	14.0360	7.4399
Improvement(%)	—	—	—	1.08	28.83	28.28	1.08	26.36	26.79
ACK (Het)	0.944	3.5704	4.1348	0.8125	23.3902	14.1663	0.767	25.8457	14.1571
SACK (Het)	0.938	3.5240	4.0116	0.9136	15.8303	10.2315	0.857	19.1993	10.7714
Improvement(%)	—	—	—	12.4487	32.3208	27.7758	11.7340	25.7157	23.9152
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
ACK (Hom)	0.945	5.2864	8.7493	0.9788	31.4170	19.6121	0.925	34.6804	19.9445
SACK (Hom)	0.945	4.4206	6.3285	0.9894	23.7265	16.2619	0.935	27.0219	17.4024
Improvement(%)	—	—	—	1.08	24.48	17.08	1.08	22.08	12.75
ACK (Het)	0.944	7.6806	11.7925	0.8125	41.7797	25.0877	0.767	46.4363	24.9903
SACK (Het)	0.938	7.7265	11.5931	0.9136	30.9080	21.7988	0.857	35.9959	22.8470
Improvement(%)	—	—	—	12.4487	26.0215	13.1096	11.7340	22.4833	8.5765

Table 2: G-ACK over ACK

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
ACK (Hom)	1.0	1.0981	0.4589	0.998	13.9772	15.6054	0.998	15.0762	15.7103
G-ACK (Hom)	0.999	1.1136	0.4240	1.0	5.5053	5.6731	0.999	6.6189	5.7434
Improvement(%)	—	—	—	—	60.61	63.65	—	61.90	63.44
ACK (Het)	1.0	1.2907	0.6591	0.956	18.7143	21.7525	0.956	19.9837	21.8786
G-ACK (Het)	1.0	1.1771	1.6585	1.0	6.4311	7.3689	1.0	7.6082	7.5607
Improvement(%)	—	—	—	4.6025	65.6354	66.1239	4.6025	61.9280	65.4425
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
ACK (Hom)	1.0	1.0785	0.3055	0.991	14.0118	9.2984	0.991	15.0916	9.3387
G-ACK (Hom)	0.999	1.1005	0.3123	1.0	5.4104	4.3494	0.999	6.5109	4.4214
Improvement(%)	—	—	—	0.91	61.39	53.22	0.81	56.86	77.14
ACK (Het)	0.998	1.2297	0.4681	0.876	18.4233	12.3731	0.874	19.6348	12.4160
G-ACK (Het)	1.0	1.1966	0.7055	1.0	6.5425	6.3140	1.0	7.7391	6.3835
Improvement(%)	—	—	—	14.1553	64.4879	48.9699	14.4165	60.5848	48.5865
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
ACK (Hom)	1.0	1.3965	0.4192	0.991	27.5326	19.7823	0.991	28.7424	19.8375
G-ACK (Hom)	0.999	1.4082	0.4191	1.0	7.7784	6.2427	0.999	8.8783	6.2779
Improvement(%)	—	—	—	0.91	71.75	68.44	0.81	69.11	68.35
ACK (Het)	0.998	1.6129	0.8443	0.876	36.8308	24.9567	0.874	38.1744	24.9986
G-ACK (Het)	1.0	1.5638	1.7119	1.0	9.0530	8.5641	1.0	10.2629	8.6914
Improvement(%)	—	—	—	14.1553	75.4200	65.6842	14.4165	73.1158	65.2325

Table 3: G-SACK over SACK (Single Destination)

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
SACK (Hom)	0.985	2.2712	3.9634	0.9969	12.2160	13.8255	0.982	14.4364	14.8146
G-SACK (Hom)	0.981	2.2204	4.7291	1.0	7.7169	7.2466	0.981	10.2697	8.9624
Improvement(%)	—	—	—	—	36.83	47.59	—	28.86	39.50
SACK (Het)	0.968	3.3283	6.3234	0.9773	16.0768	18.6827	0.946	19.2518	19.7812
G-SACK (Het)	0.949	3.6595	8.0954	0.9989	10.6899	11.9077	0.948	14.2612	14.3889
Improvement(%)	—	—	—	2.2178	33.5073	36.2635	0.2114	25.9228	27.2597
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
SACK (Hom)	0.945	2.2197	2.1982	0.9894	11.8276	6.9466	0.935	14.0360	7.4399
G-SACK (Hom)	0.934	2.2778	2.4718	1.0	7.5037	4.2978	0.934	9.9727	5.5487
Improvement(%)	—	—	—	—	36.56	38.13	—	28.95	25.42
SACK (Het)	0.938	3.5240	4.0116	0.9136	15.8303	10.2315	0.857	19.1993	10.7714
G-SACK (Het)	0.909	3.3011	3.9439	0.9978	9.8746	6.8912	0.907	13.1107	8.2316
Improvement(%)	—	—	—	9.2108	37.6222	32.6472	5.8343	31.7126	23.5791
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
SACK (Hom)	0.945	4.4206	6.3285	0.9894	23.7265	16.2619	0.935	27.0219	17.4024
G-SACK (Hom)	0.934	4.6034	7.2524	1.0	12.9835	8.3755	0.934	16.7482	11.0858
Improvement(%)	—	—	—	—	45.28	48.50	—	38.02	36.30
SACK (Het)	0.938	7.7265	11.5931	0.9136	30.9080	21.7988	0.857	35.9959	22.8470
G-SACK (Het)	0.909	7.2436	11.1326	0.9978	18.1307	13.7957	0.907	22.8654	15.9647
Improvement(%)	—	—	—	9.2108	41.3398	36.7135	5.8343	36.4778	30.1234

Table 4: G-SACK over SACK (Multiple Destinations)

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
SACK (Hom)	0.970	2.1481	3.6806	0.999	11.3114	12.9743	0.969	13.4591	14.0446
G-SACK (Hom)	0.968	2.3295	4.2461	1.0	3.6124	2.4851	0.968	5.9419	5.1368
Improvement(%)	—	—	—	—	68.06	80.85	—	55.85	63.43
SACK (Het)	0.942	3.2730	6.7176	0.963	16.7625	19.5349	0.907	19.7835	20.5188
G-SACK (Het)	0.941	2.7197	5.3218	1.0	4.1776	3.4813	1.0	6.8973	6.5647
Improvement(%)	—	—	—	3.8589	75.0777	82.1791	3.7486	65.1361	68.0064
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
SACK (Hom)	0.835	2.0993	1.2617	0.9748	11.8657	4.0935	0.814	13.9587	4.4508
G-SACK (Hom)	0.82	2.2526	1.3508	1.0	3.5580	1.3150	0.82	5.8106	2.0040
Improvement(%)	—	—	—	2.59	70.01	67.88	0.74	58.37	54.97
SACK (Het)	0.781	2.880	2.1515	0.738	16.2845	5.7566	0.576	18.9963	6.0265
G-SACK (Het)	0.789	2.5933	1.8854	1.0	4.1386	1.6928	0.789	6.7319	2.6242
Improvement(%)	—	—	—	35.5903	74.5856	70.5886	36.9792	64.5620	56.4557
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
SACK (Hom)	0.835	6.7624	7.5608	0.9748	37.7574	18.1125	0.814	41.3973	18.9178
G-SACK (Hom)	0.82	7.7116	8.6348	1.00	6.9207	2.7467	0.82	12.5310	8.7067
Improvement(%)	—	—	—	2.59	81.67	84.84	0.74	69.73	53.98
SACK (Het)	0.781	11.0969	14.4008	0.738	51.1982	20.9116	0.576	55.9285	21.2177
G-SACK (Het)	0.789	9.5838	12.1490	1.0	8.3065	3.9326	0.789	15.2749	11.8932
Improvement(%)	—	—	—	35.5903	83.7758	81.1942	36.9792	72.6885	43.9468

Table 5: Benefits of coding: 3 unicast flows.

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
w/o coding (Hom)	0.981	2.2295	3.4012	0.994	16.0823	16.1756	0.975	18.3038	16.8181
coding (Hom)	1.0	1.2229	0.5248	0.991	15.7965	16.7567	0.991	17.0185	16.8304
Improvement (%)	1.94	45.15	84.57	—	—	—	—	—	—
w/o coding (Het)	0.955	3.5933	7.7669	0.9340	21.6663	22.5725	0.892	25.1213	23.4468
coding (Het)	1.0	1.3460	0.5940	0.9340	20.1394	22.4994	0.934	21.4770	22.6113
Improvement(%)	4.7120	62.5414	92.3522	—	—	—	—	—	—
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
w/o coding (Hom)	0.937	2.4574	2.5864	0.9785	16.8021	10.2819	0.917	19.2071	10.5719
coding (Hom)	1.0	1.2272	0.2863	0.982	16.3564	9.5734	0.982	17.5824	9.6066
Improvement (%)	6.72	50.06	88.93	—	—	—	—	—	—
w/o coding (Het)	0.9280	2.8425	3.3681	0.8341	21.0305	13.0562	0.7740	23.8021	13.4329
coding (Het)	1.0	1.3460	0.3179	0.8340	22.4950	13.2082	0.8340	23.7822	13.2330
Improvement(%)	7.7586	54.4169	90.5614	—	—	—	—	—	—
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
w/o coding (Hom)	0.937	5.0969	7.3967	0.9785	31.3528	19.8724	0.917	34.5955	20.1253
coding (Hom)	1.0	1.6805	0.4183	0.982	30.5181	19.2508	0.982	31.8224	19.2787
Improvement (%)	6.72	67.03	94.34	—	—	—	—	—	—
w/o coding (Het)	0.9280	6.0617	9.5697	0.8341	40.9468	24.9813	0.7740	44.4334	25.1125
coding (Het)	1.0	1.7814	0.4769	0.8340	43.5711	24.8722	0.8340	44.9707	24.9106
Improvement(%)	7.7586	70.6122	95.0166	—	—	—	—	—	—

Table 6: Benefits of coding: 10 unicast flows.

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
w/o coding (Hom)	0.923	5.4488	9.3091	0.986	18.4437	17.0452	0.911	23.6602	19.0294
coding (Hom)	1	2.3560	0.9330	0.990	19.6468	18.3935	0.990	22.0005	18.4809
Improvement (%)	8.34	56.7611	89.98	—	—	—	—	—	—
w/o coding (Het)	0.8710	7.6130	13.5211	0.9392	21.1970	21.1498	0.8180	28.1932	23.3666
coding (Het)	1	2.5549	0.9575	0.9250	24.8490	22.0173	0.9250	27.2924	22.8159
Improvement (%)	14.8106	66.4403	92.9185	—	—	—	—	—	—
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
w/o coding (Hom)	0.431	5.7216	2.7548	0.925	18.1435	5.3114	0.431	23.7155	5.8263
coding (Hom)	1	2.3666	0.2765	0.9280	18.7219	5.3958	0.9280	21.0892	5.4245
Improvement (%)	132.02	58.6374	89.9630	—	—	—	—	—	—
w/o coding (HeT)	0.345	7.0914	3.8001	0.5311	20.3529	6.8386	0.1880	26.8281	7.0299
coding (HeT)	1	2.8503	0.3443	0.4660	24.1425	7.1262	0.466	26.9835	7.1489
Improvement (%)	182.4859	59.8062	90.9397	—	—	—	—	—	—
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
w/o coding (Hom)	0.431	24.9669	18.2471	0.925	51.0408	18.3003	0.431	60.2884	17.0522
coding (Hom)	1	3.8020	0.6036	0.928	52.4117	17.8778	0.9280	54.8088	17.9445
Improvement (%)	132.02	84.77	96.69	—	—	—	—	—	—
w/o coding (Het)	0.345	32.1189	22.8868	0.5311	62.7478	20.1436	0.1880	71.5479	18.1023
coding (Het)	1	4.6869	0.7680	0.4660	68.2396	18.1035	0.466	71.1448	18.0804
Improvement (%)	182.4859	85.4077	96.6444	—	—	—	—	—	—

Table 7: Proposed over Basic Scheme: Topology 3

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
Basic (Hom)	0.981	2.2295	3.4012	0.994	16.0823	16.1756	0.975	18.3038	16.8181
Proposed (Hom)	1.0	1.2661	0.5103	1.0	7.7358	7.0283	1.0	9.0018	7.1703
Improvement(%)	1.94	43.21	85.0	0.62	51.90	56.55	2.56	50.82	57.37
Basic (Het)	0.955	3.5933	7.7669	0.934	21.6663	22.5725	0.892	25.1213	23.4468
Proposed (Het)	1.0	1.5811	0.6558	0.996	16.4786	15.6688	0.996	19.6437	15.8713
Improvement(%)	4.7120	55.9987	91.5565	6.6345	23.9436	30.5846	11.6592	21.8046	32.3093
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
Basic (Hom)	0.937	2.4574	2.5864	0.9785	16.8021	10.2819	0.917	19.2071	10.5719
Proposed (Hom)	1.0	1.2639	0.2951	1.0	7.8367	4.5957	1.0	9.1006	4.6528
Improvement(%)	6.72	48.571	88.59	2.20	53.36	55.30	9.05	52.62	55.99
Basic (Het)	0.928	2.8425	3.3681	0.8341	21.0305	13.0562	0.774	23.8021	13.4329
Proposed (Het)	1.0	1.4663	0.3315	0.990	14.5694	9.6351	0.990	17.4992	9.7775
Improvement(%)	7.7586	48.4151	90.1577	18.6977	30.7225	26.2029	27.9070	26.4804	27.2123
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
Basic (Hom)	0.937	5.0969	7.3967	0.9785	31.3528	19.8724	0.917	34.5955	20.1253
Proposed (Hom)	1.0	1.7284	0.4293	1.0	13.8886	9.0026	1.0	15.3496	9.0492
Improvement(%)	6.72	66.09	94.20	2.20	55.70	54.70	9.05	55.63	55.04
Basic (Het)	0.928	6.0617	9.5697	0.8341	40.9468	24.9813	0.774	44.4334	25.1125
Proposed (Het)	1.0	2.0298	0.5399	0.990	25.4897	17.3482	0.990	28.6676	17.3814
Improvement(%)	7.7586	66.5143	94.3582	18.6977	37.7492	30.5553	27.9070	35.4594	30.7859

Table 8: Proposed over Basic Scheme: Topology 4

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
Basic (Hom)	0.952	2.4497	4.1897	0.9936	16.6622	17.2934	0.946	19.0762	18.1168
Proposed (Hom)	1.0	1.2695	0.5459	1.0	5.1643	4.9591	1.0	6.4338	5.1240
Improvement(%)	5.01	48.18	86.97	0.64	69.01	71.32	5.71	66.27	71.72
Basic (Het)	0.939	3.2399	5.3714	0.9297	22.2170	22.5343	0.873	25.3847	23.0115
Proposed (Het)	1.0	1.3104	0.5417	1.0	10.7862	9.7979	1.0	12.0955	9.8617
Improvement(%)	6.4963	59.5543	89.9151	7.5601	51.4507	56.5201	14.5475	52.3512	57.1445
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
Basic (Hom)	0.867	2.2018	1.5887	0.9584	17.1848	7.0010	0.831	19.3647	7.2744
Proposed (Hom)	1.0	1.2588	0.2648	1.0	4.1086	2.5586	1.0	5.2783	2.6651
Improvement(%)	15.34	42.83	83.33	4.34	76.09	63.45	20.34	72.74	63.36
Basic (Het)	0.839	3.0645	2.8543	0.6555	22.4220	9.4315	0.550	25.4417	9.7270
Proposed (Het)	1.0	1.4803	0.3286	1.0	9.5957	6.0071	1.0	11.0750	6.0992
Improvement(%)	19.1895	51.6952	88.4875	52.5455	57.2041	36.3081	81.8182	56.4691	37.2962
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
Basic (Hom)	0.867	6.0504	7.3123	0.9584	42.8873	19.2729	0.831	45.7103	19.6193
Proposed (Hom)	1.0	1.8318	0.4293	1.0	8.9410	5.8355	1.0	10.3393	5.9502
Improvement(%)	15.34	69.72	94.13	4.34	79.15	69.72	20.34	77.38	69.67
Basic (Het)	0.839	9.6924	13.5074	0.6555	56.0690	21.8054	0.550	60.0094	21.8054
Proposed (Het)	1.0	2.1560	0.5549	1.0	19.0522	12.8936	1.0	20.6085	12.9333
Improvement(%)	19.1895	77.7558	95.8919	52.5455	66.0201	40.8697	81.8182	65.6579	40.6876

Table 9: Proposed over Basic Scheme: Topology 5

	Forward			Return			Round Trip		
Scheme	P_{11}^f	$E[D_{11}^f]$	$\sigma[D_{11}^f]$	P_{11}^r	$E[D_{11}^r]$	$\sigma[D_{11}^r]$	P_{11}^{rt}	$E[D_{11}^{rt}]$	$\sigma[D_{11}^{rt}]$
Basic (Hom)	0.963	2.3487	3.7792	0.9958	16.6338	17.0530	0.959	18.9828	17.5904
Proposed (Hom)	1.0	1.2933	0.5621	1.0	3.9447	3.6940	1.0	5.2380	3.8859
Improvement(%)	3.84	44.94	85.13	0.42	76.29	78.34	4.28	72.41	77.91
Basic (Het)	0.957	2.2641	4.5065	0.9373	19.5794	20.8895	0.8970	22.0095	21.3864
Proposed (Het)	1.0	1.2189	0.5717	1.0	5.7374	5.2396	1.0	6.9562	5.3407
Improvement(%)	4.4932	46.1640	87.3139	6.6890	70.6968	74.9175	11.4827	68.3946	75.0276
Scheme	P^f	$E[D_{avg}^f]$	$\sigma[D_{avg}^f]$	P^r	$E[D_{avg}^r]$	$\sigma[D_{avg}^r]$	P^{rt}	$E[D_{avg}^{rt}]$	$\sigma[D_{avg}^{rt}]$
Basic (Hom)	0.829	2.3312	1.6827	0.9565	16.4299	5.5921	0.793	18.7245	5.8749
Proposed (Hom)	1.0	1.2746	0.2493	1.0	3.9598	2.1756	1.0	5.1647	2.2658
Improvement(%)	20.63	45.32	85.18	4.55	75.90	61.10	26.10	72.42	61.43
Basic (Het)	0.768	2.7840	2.0624	0.5195	21.9157	7.1348	0.3990	25.5429	7.2732
Proposed (Het)	1.0	1.2817	0.2518	1.0	6.1006	3.6782	1.0	7.3818	3.7186
Improvement(%)	30.2083	53.9619	87.7909	92.4812	72.1633	48.4470	150.6266	69.9229	48.8726
Scheme	P^f	$E[D_{max}^f]$	$\sigma[D_{max}^f]$	P^r	$E[D_{max}^r]$	$\sigma[D_{max}^r]$	P^{rt}	$E[D_{max}^{rt}]$	$\sigma[D_{max}^{rt}]$
Basic (Hom)	0.829	8.5356	11.4188	0.9565	47.1655	18.0528	0.793	50.7366	18.4644
Proposed(Hom)	1.0	1.9503	0.4802	1.0	8.0163	4.4875	1.0	9.0674	4.5419
Improvement(%)	20.63	77.15	95.79	4.55	83.0	75.14	26.10	82.13	75.40
Basic (Het)	0.768	10.4635	13.1736	0.5195	62.9966	20.2241	0.3990	66.2896	20.0738
Proposed (Het)	1.0	2.0289	0.5491	1.0	12.8768	8.7825	1.0	14.3392	8.7896
Improvement(%)	30.2083	80.6979	95.8318	92.4812	79.5595	56.5741	150.6266	78.3689	56.2136

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. H is presently a final year Ph.D. candidate in Information Technology, Telecommunications and Electronics jointly with Institute of Telecom SudParis and UPMC (Paris VI). Between 2005 and 2008, he served as an Audit Officer in the office of Auditor General of Pakistan. His current research interests are in the areas of delay/disruption tolerant networks, wireless mobile ad hoc networks and network coding.

Eitan Altman Eitan Altman received the B.Sc. degree in electrical engineering (1984), the B.A. degree in physics (1984) and the Ph.D. degree in electrical engineering (1990), all from the Technion-Israel Institute, Haifa. In (1990) he further received his B.Mus. degree in music composition in Tel-Aviv university. Since 1990, Dr. Altman has been a researcher at INRIA (National research institute in computer science and control) in Sophia-Antipolis, France. He has been in the editorial boards of several scientific journals: Wireless Networks (WINET), Computer Networks (COMNET), Computer Communications (Comcom), J. Discrete Event Dynamic Systems (JDEDS), SIAM J. of Control and Optimisation (SICON), Stochastic Models, and Journal of Economy Dynamic and Control (JEDC). He received the best paper award in the Networking 2006, in Globecom 2007 and in IFIP Wireless Days 2009 conferences, and is a coauthor of two papers that have received the best student paper awards (at QoFis 2000 and at Networking 2002). His areas of interest include networking, stochastic control and game theory. More informaion can be found at www-sop.inria.fr/members/Eitan.Altman/

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 Professor in the Networks and Services department in the Telecom SudParis, France. His research interests are in the area of quality of service and teletraffic engineering, notably in wireless networks.

Manoj Panda (M.Tech. (EE), IIT Kanpur and Ph.D. (ECE), IISc Bangalore) is a post-doctoral researcher at Telecom SudParis. Previously he was a post-doctoral researcher at INRIA Sophia Antipolis (MAESTRO Group). His area of research is performance modeling of communication networks.

arshad.jpg
Click here to download high resolution image

mage

Click here to download high resolution image

manoj-kumar-photo.jpg
Click here to download high resolution image