Network Coding Design and Testbed Validation for Reliable and Secure Satellite Overlay Networking

M. A. Vázquez-Castro, Tan Do-Duy, P. Saxena, F. Vamstad, H. Skinnemoen, (TBC I. Panagiotopoulos)

***Abstract*—In this paper, we present a novel functional design and testbed validation of network coding (NC) for reliable and secure overlay networking. We target innovative scenarios requiring flexible and configurable topological and functional design. For example, our scenarios include but are not limited to line toplogies involving tactical drones for visual situational awareness. For such scenarios, while satellite communications can secure connectivity beyond the reach of terrestrial networks and line of sight datalinks, NC can secure reliability and security.**

Our contribution is three-fold. First, we describe in detail our proposed functional design that enables flexible instantiation of different types of network codes via a set of configurable network coding (C-NC) functions. The choice of coefficients and overall NC scheme depends on the performance target, which includes reliability and security. We also present our novel construction of structured network codes based on Pascal matrices, which can be deterministic or randomized (thus enabling also security in addition to reliability). Second, we describe our testbed developed under the SatNetCode research project funded by the European Space Agency. In order to validate the experimental performance of our overall design, we make use of an innovative optimization procedure to identify the optimal coding rate to achieve a target service quality. Finally, we present our experimental results obtained for the SATCOM realistic case of overlay networking with limited bandwidth over a GEO satellite. Our results show the usefulness of our analytical models and the benefits of our overall design. In particular, our results show the good structural properties and delay-reliability tradeoffs of our novel network codes using Pascal matrices, which can be a good alternative to random network codes in some SATCOM use-cases.

***Index Terms*—Network coding, Pascal matrices, functional architecture, reliability-delay trade-off.**

I. Motivation and Contributions

A. Motivation and scenarios of interest

NETWORK coding is by now a well-known technology first proposed in [1]. It has been already implemented in real systems to improve different metrics of network performance, such as reliability, security or resilience against link failures. For satellite applications, different solutions and implementations have been proposed to date, see e.g. [2] and references therein. In this case, it is known that the impact of satellite links on NC performance imposes specific design constraints (e.g. [2] [3]). Here, we tackle the case of using network coding to provide reliability and security to (heterogeneous) overlay networks that include either UAVs, satellite links or both.

We target innovative scenarios requiring flexible and configurable topological and functional design. For example, our scenarios include but are not limited to line topologies involving tactical drones to provide reliable and secure transmission for visual situational awareness [4]. For such scenarios, while satellite communications can secure connectivity beyond the reach of terrestrial networks and line of sight datalinks, NC can secure reliability and security. An illustration of an example of target use-case is shown in Fig. 1.

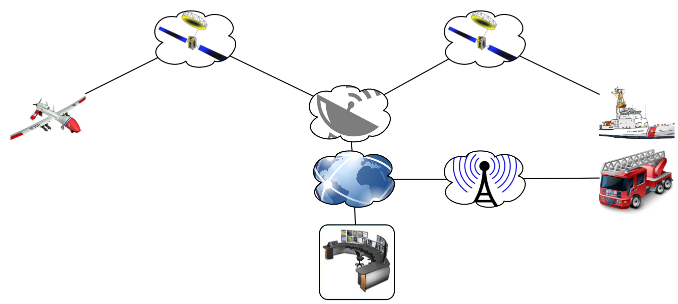


Fig. 1: Example of target use-case with three nodes. First link is the satellite link. Second link can be WiFi or Satellite links.

UAVs offer safe operations, improved data and lower cost operations. It is relevant to note that while piloting these vehicles beyond light of sight require high reliability and low delay, payload users often require reliability and security but low operational cost. Hence, when the latter case is considered as a service from the application developer perspective, the implied technical constraint for packet-level networking design is low bandwidth and if possible, inherent security in packet transmission. Further, it is also possible to assume that potential users will accept higher delay but lower cost and higher robustness and reliability. Hence, delay-reliability trade-off arise as a relevant performance metric in our target use-case scenarios.

Clearly, NC is an enabling technology for the above use-cases which are clearly different to traditional use-cases such as video streaming. Therefore from the user point of view, we can identify the following objectives to consider when addressing the design of NC solutions:

1) Bandwidth Saving: the overall solution should take into account the operational cost, specially when expensive satellite resources are used and employ strategies of bandwidth saving and cost reduction.

Ángeles Vazquez-Castro and Tan Do-Duy are with the Department of Telecommunications and Systems Engineering, Autonomous University of Barcelona, Barcelona, Spain e-mail: angeles.vazquez, tan.doduy@uab.es.

P. Saxena is with the Department of Computer Science, BITS Pilani University, Hyderabad, India, e-mail: psaxena@hyderabad.bits-pilani.ac.in.

F. Vamstad, H. Skinnemoen are with AnsuR Technologies, Oslo, Norway. e-mail:fredrick, harald @ansur.no

2) Robustness: network codes should provide robustness and be able to support different networks for continuous connectivity (e.g. for tactical operations). Networks including WiFi, Internet, Cellular, TETRA and satellite hops can be used to provide continuous, global and rapid connectivity.
3) Reliability and security: different codes should be available to counteract unstable and varying network conditions with packet erasures. The different codes can provide different quality of service targets defined in terms of acceptable packet erasure rate and in-order packet delay after decoding. The different codes may also provide inherent security due to the several options to choose the network coding coefficients.

In order to use NC technology for such non traditional use-cases there is the need of novel design frameworks. In particular, such frameworks should be flexible to facilitate rapid network topology and functionalities' configurability. For this reason, we have developed a functional design that enables flexible instantiation of different types of network codes via a set of configurable network coding (C-NC) functions. The choice of coefficients and overall NC scheme depends on the performance target, which includes reliability and security. We also present our novel construction of structured network codes based on Pascal matrices, which can be deterministic or randomized (thus enabling also security in addition to reliability). Next, we summarize our contributions.

B. Main contributions

Our main contributions can be summarized as follows:

1) **Holistic system modelling and functional network coding architecture design**. We propose a novel holistic system modeling. It allows control over reconfigurable logical topology and network functionalities (aligned with software-defined networking principles). It consists of the definition of different types of logical nodes depending on their encoding, recoding and decoding functionalities and whether or not the satellite (overlay) application designer has control over them. We have also developed the functional design of network coding, re-encoding and decoding that can instantiate different network coding schemes depending on the target performance metric.
2) **Implementation of an experimental test-bed and set of network codes, including a novel code carved from Pascal matrices**. We have implemented our functional design on a testbed composed of odroids (to emulate the channel) and computers. The NC functions have a set of available coding coefficients among which we have proposed structured coefficients carved from Pascal matrices. Their algebraic structure allows allows deterministic or randomized construction, thus inherently enabling also security in addition to reliability.
3) **Experimental test-bed validation and results for realistic use-case scenarios.** The testbed software architecture can instantiate different network codes via the reconfigurable block of functions. We show our pre-analysis of structural and operational properties of three representative network codes justifying the implementation of the systematic network codes only. For performance evaluation, we make use of an optimization procedure developed by us in [10] to identify the optimal coding rate to achieve a target service quality. Our results show the usefulness of our analytical models and the benefits of our overall design. In particular, our results show the good structural properties and delay-reliability tradeoffs of our novel network codes using Pascal matrices, which can be a good alternative to random network codes in some SATCOM use-cases.

This paper is structured as follows. In Section II we present our holistic approach to system model and assumptions and define the selected performance metrics. In Section III we describe our proposed functional architecture design of network coding functionalities. In Section IV we describe the network codes our functional design can instantiate. In Section V we provide technical details of our testbed. This testbed has been developed during the project SatNetCode funded by the European Space Agency [4]. In Section VI we present a preliminary experimental validation for the case of line network with three nodes. Specifically, we perform an analysis of structural and operational properties of three representative types of network codes. In Section VII we identify preliminary conclusions and outline further work.

II. System Model And Performance Metrics

A. System-level holistic modelling

Our proposed novel design framework is based on the following two key observations.

First, we are addressing applications operating over-the-top (i.e. overlay) networks. Moreover, our focus is network coding for SATCOM but from the application developer perspective. Hence, our network coding design is driven by system design rather than by network design since application developers do not have access to network kernel. Hence, our underlying system-level model needs to be holistic, i.e. we assume network coded flows traverse satellite links and wireless links with different bandwidths, erasure rates and delays. Further, available satellite links are characterized with low bit rates, long delays and expensive billing. Furthermore, aside from heterogeneous links, also physical devices involved in the communication process are assumed having heterogeneous size, computational capabilities, etc. Hence, the holistic nature of the system model requires the design of highly flexible and configurable network coding solutions to control the reliability of information flows. We note that our system-driven design contrasts with the usual design approach mostly considered in developing network coding applications, where low complexity and highest encoding/decoding efficiency are main design targets. To see this note that, for example, a satellite terminal may afford high computational complexity in order to better use expensive bandwidth.

Second, we assume centralized control rather than distributed operation. Specifically, we assume an overlay operational system that runs over the top of traditional networks (WiFi, cellular, etc.). In order to operate such system, the application developer aims for full control to instantiate different network coding operations for better end-user experience. The full control is enabled by remote control of network coding (virtual or physical) nodes as we propose here. Hence, our design is centralised which enables easy control and reconfiguration of network coding nodes and codes.

Our baseline network-level assumption is hybrid best effort transmission over overlay networks. Hence, network coding is used to control selected metrics to optimize reliable transmission according to target quality metrics. We define the following types of logical nodes:

- **Source node**, which we denote as SOU and runs the encoding functionalities.
- **Re-encoding node**, which can be of two types: Relay node, which we denote as REL or intermediate node, which we denote as INM. The difference between REL and INM types of nodes is that the network coding operation of nodes REL are under the control of the satellite application designer while the operation of nodes INM are under the control of the connection service operators. In either case, these nodes run re-encoding functionalities.
- **Decoding node**, which we denote as DES and are the sink nodes running the decoding functionalities.

We assume per-flow line (multi-hop) network logical topology of n nodes. There may be multicast transmission from the last line network node to n_{mul} nodes. Link heterogeneity can be described with Markovian models, the simplest being a Gilbert channel model per ℓ-th link represented by the transition matrix, H^ℓ, with elements representing the probabilities of transitioning between the two channel states, the *good* state and the *bad* state as

$$H^\ell = \begin{bmatrix} 1-g^\ell & g^\ell \\ b^\ell & 1-g^\ell \end{bmatrix}.$$

The steady state probabilities for the bad channel conditions are $\pi_g^\ell = g/(g+b)$ and $\pi_b^\ell = b/(g+b)$. Hence the erasure probability for ℓ-th link is $\epsilon^\ell = \pi_b^\ell$ and the model can be completely characterized by ϵ^ℓ and the average erasure burst length $D_b^\ell = 1/g$. For a general per-flow line network topology of n nodes with per-link erasure ϵ^ℓ, $1 \le \ell \le n-1$ the capacity is given as [5]

$$\mathcal{C} = \min_{1\le\ell\le n-1}(1-\epsilon^\ell), \tag{1}$$

B. Definition of Performance Metrics

We distinguish performance metrics depending on whether they describe the structural properties of the network code or they describe the operational properties. The following metrics describe the structural properties. Our definitions are coherent with related literature [3][4].

Definition 1. ***Coding Redundancy,*** *R. It is defined [3] as the total amount of redundancy added by the network code at the source.*

Definition 2. ***Packet efficiency,*** *η^{pkt}. It is defined [3] as the total number of DoFs (i.e., the total number of information packets) that need to be transferred divided by the actual number of packets (both uncoded and coded) received by the sink.*

Definition 3. ***DoF efficiency,*** *η^{DoF}. It is defined as the total number of DoFs (i.e., the total number of information packets) that need to be transferred divided by the number of innovative packets received by the sink.*

Note that both definitions account for structural properties: the first one measures the efficiency in terms of overall induced traffic flow while the second one only in terms of effective (innovative) traffic.

The following (per-destination) metrics describe the operational properties, i.e. they are directly related to application performance.

Definition 4. ***In-order average delivery,*** *$\bar{D}$. It is the average of the difference between the time an information packet is first transmitted and the time that the same packet is delivered, in-order to the destination.*

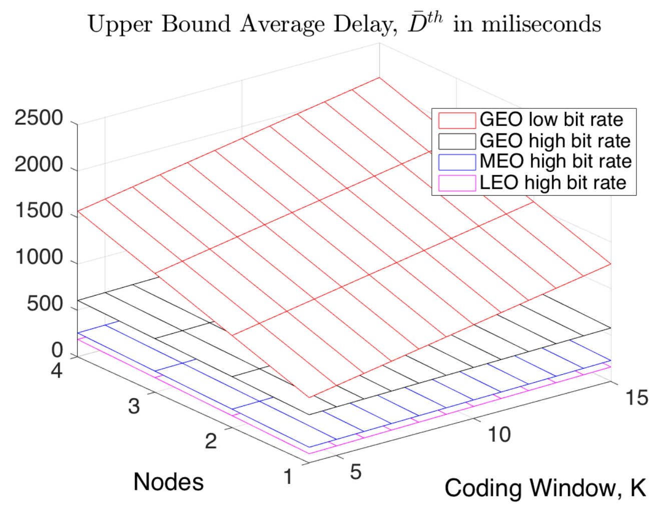


Fig. 2: Theoretical upperbounds for different orbit and rate assumptions and symmetric case of per-link 5% erasures.

The theoretical average delay, $\bar{\mathcal{D}}_n^{th}$, assuming a unique worst link $p_m = \max_{1\le\ell\le n-1}\epsilon^\ell$ is upper bounded as [5]

$$\bar{\mathcal{D}}_n^{th} \le \frac{n}{1-p_m} + \sum_{i=1, i\ne m} \frac{p_m}{p_m - p_i}. \tag{2}$$

Fig. 2 shows numerical values of the theoretical delay upper bound (2) for different orbit and rate assumptions. The plot assumes satellite hops with corresponding expected latencies, it illustrates the known effect of the orbit but also of the available bandwidth (high bit rate is 2.5 Mbps and low bit rate is 250 Kbps).

Definition 5. ***Average packet erasure rate after decoding,*** *$\bar{\eta}$. It is defined as the average packet erasure rate that is achieved at the destination after the overall coding and decoding operations.*

Note that we need to consider average (residual) packet erasure rate after decoding because with satellite links, we do

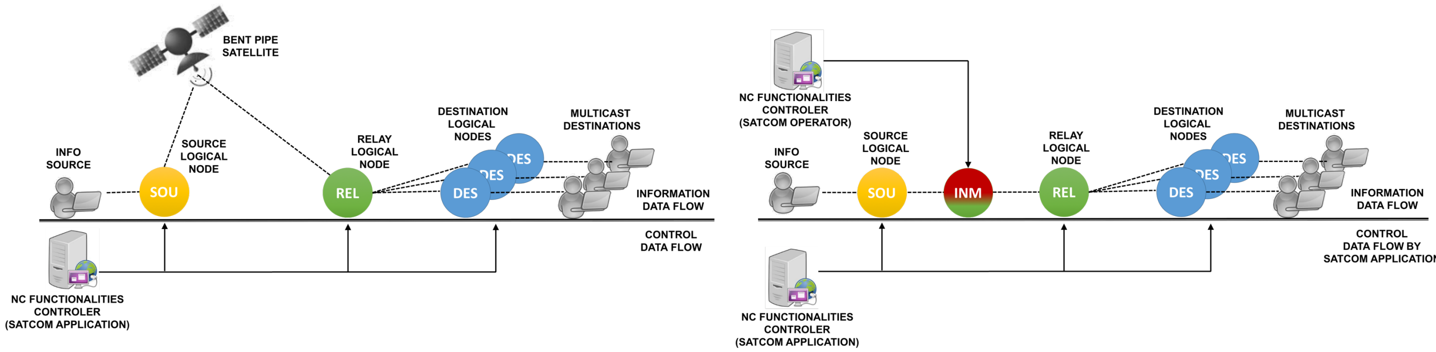


Fig. 3: Example of simple single relay NC topology (left) and two recoders (right) with NC controllers. Black arrows indicate data flows and dashed arrows control flows.

not assume feedback to achieve zero errors (with e.g. rateless coding).

The theoretical average packet erasure rate is known for typical random network codes and we have derived theoretical performance for our proposed structured codes, see Section IV below.

III. Functional Architecture

We have designed a per-node functional architecture based on identifying common sub-functionalities to any network coding scheme. The idea is that such common operational sub-functionalities are in one-to-one correspondence with efficient and well defined software modules, one of which is the network coding library. We allow the functional NC parameters to adapt to network statistics and functional NC policies to be loaded or reconfigured by a centralized network coding service controller. For illustration, a schematic view of the controller operation is shown in Fig. 2 for a simple single relay NC topology. The controller also selects and assigns the logical role of the nodes and can load or reconfigure functional parameters and policies for the NC functionalities running at the nodes. The selection of physical nodes as REL or INM logical nodes depends on the scenario and performance objectives. Here we assume that they are strategically chosen by the appropriate logical or physical entities. Such optimal selection will often result in a small number of nodes the packet flow traverses, which should suffice to optimize a selected performance metric of interest. The design of the controller has been left out of the scope of the project, ongoing work is partly described in [6]. Here, we assume that input packet flows are given as an input flow for intra-session network coding. For our validation tests, we will be given the topology of nodes along with their logical role. Figs. 4-6 show the high-level functional description of the encoding, re-encoding and decoding functions. The centralized controller to drive coding or to orchestrate the configuration of the functional parameters and policies is not shown here. A high-level description of the internal logic of our designed functionalities is now given as follows. The internal logic is controlled and we distinguish for convenience the control data flow for software control and for coding control.

The *Encoder functional architecture* is shown in Fig. 3. It takes the input flow, which is first segmented into source

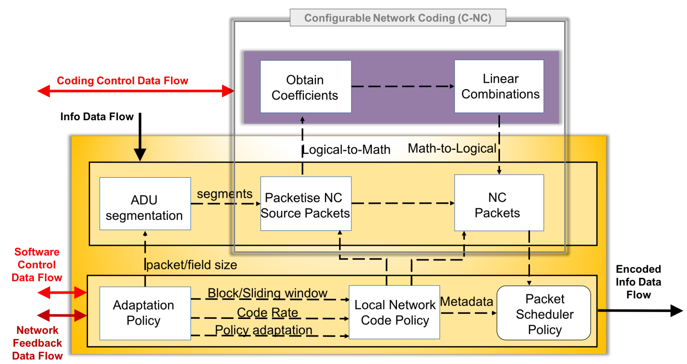


Fig. 4: Encoder functional architecture.

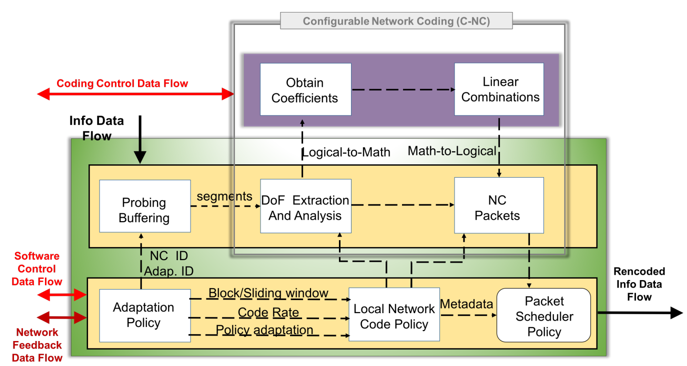


Fig. 5: Re-encoder functional architecture.

packets, whose size can be controlled. These packets are then mathematically interpreted and transformed according to all the inputs that are given. The transformed packets are then logically interpreted as repair packets. The set of functions generating coding coefficients and performing algebraic combinations is a configurable network coding (C-NC) set of functions. Reconfiguration may also mean different mathematical interpretation of the physical digital payload. For this reason, the mathematical interpretation is explicitly shown. Note that the C-NC can receive coding coefficients from local resources or remote centralized resources.

The Adaptation Policy is the functional part of the internal logic in charge of the adaptation to network dynamics, from

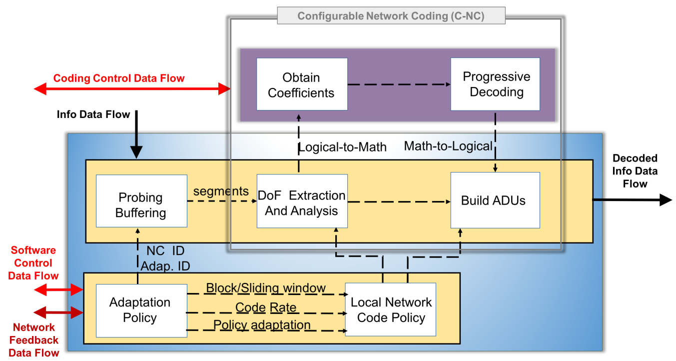


Fig. 6: Decoder functional architecture.

which periodically receives feedback, for example, on the received degrees of freedom (DoF) from the destination (DES) nodes or network state statistics. The codebook manager may keep locally a selcted set of coefficients that the function can apply and can be updated and/or reconfigured. The DoF scheduler distributes the physical DoFs along the time domain (e.g. for reliability) or the space domain (e.g. for caching). Schedulers can be fine-grained optimized to provide full control over different performance metrics. Our scheduler generalizes our ideas in [8], whereby recoding nodes do not apply random linear combinations, but some structured scheduling of available DoFs.

The *Re-encoder functional architecture* is shown in Fig. 4. It can run at the REL or INM nodes. We assume for now that the internal logic of the recoding functionalities is the same for both cases as follows. The input flow is probed to identify the IDs of the several choices taken at the source and stored in buffers of optimized size. The rank of the set of packets at the node is computed and the DoFs are mathematically interpreted and transformed according to the C-NC functions. The available DoFs at the node are scheduled for forwarding according to the selected policy to produce the re-encoded flow.

The *Decoder functional architecture* is shown in Fig. 5. It runs at the DES nodes. The input flow is probed to identify the IDs of the several choices taken at the source and stored in buffers of some optimal size. The rank of the set of packets at the node are then decoded progressively, using different algorithms for Gaussian elimination.

IV. Network Codes

Our architecture can instantiate different network codes via the reconfigurable block of functions C-NC. Here, we analyze the performance of three representative NCs. We choose them with fixed-size coding window but instantiations with sliding window are also possible.

A. Codes with no structured (random) coefficients

It is known that random linear network coding (RLNC) achieves capacity with high probability for large field and packet sizes. However, the main drawback of RLNC is its complexity since for fixed block coding size of K packets, encoding complexity scales as $\mathcal{O}(K^2 L_{pkt})$ while decoding as $\mathcal{O}(K^3+K^2 L_{pkt})$ (through Gaussian elimination). For realistic finite-length transmission, the tradeoff between alphabet size, packet length, and overhead must be carefully weighed. Hence we consider as baseline systematic linear network coding (SLNC), which has been shown to have near-optimal performance with lower delay and complexity [7]. In this case, the K packets are sent uncoded while $(KR-K)$ packets are sent coded. Our baseline SLNC sends systematic packets first and encoded packets after, i.e., regular SLNC with no optimized scheduling of DoFs is assumed for this case.

Our architecture can also generate DoF-aided random encoding (DARE), which are codes that can be seen as modified SLNC codes that use a smart scheduler of DoFs at recoding nodes. The choice of scheduling policy depends on the target performance metric. Here we implement our DARE code as described in [8].

For both SLNC and DARE, the SOU node encodes the source packets with a random $K \times KR$ matrix G_1 with entries in $\mathbb{F}_q$. Assuming KR uses of the channel, the line network channel with SLNC and SOU is described as

$$Y_n = \prod_{\ell=2}^{n-1} G_\ell D_{\epsilon_\ell} G_1 X = W_n X_{coded}, \tag{3}$$

where

$$X_{coded} = [\mathbb{I}_{K\times KR} \quad G_1 S_{(L_{pkt}-K)\times K}],$$

and G_ℓ are the recoding matrices corresponding to SLNC and DARE. D_{ϵ_ℓ} are the diagonal matrices with elements taking 0 value with probability ϵ_ℓ and 1 value with probability $1-\epsilon_\ell$. RLNC is capacity achieving for the stochastic linear operator W_n when $R > 1/(1-\mathcal{C})$. Note that other codes with random generation of coefficients for different network applications (e.g. requiring capacity-achieving properties) such as BATS codes [9] admit the same channel description.

Theoretical performance of random network codes is well known. The source sends E_i redundant packets in addition to the K information packets. The value of E_i is deterministic in the non systematic case and random in the systematic case. We denote a binomial random variable $V \sim bin(n, p_i)$ with the probability mass function $Pr(V=v) = \binom{n}{v} p_i^v (1-p_i)^{n-v}$, $v = 0, 1, ..., n$ with $p_i = 1-\epsilon_i$. For easy comparison with our derivations in the next section, we arrange the analytical expressions from [7] as follows.

For a total number $L+1$ multiple hops (L directed links), K packets are transmitted from source node 0 to destination node L. The probability that a transmitted packet is received on the link between nodes $i-1$ and i is $p_i = 1-\epsilon_i$, with ϵ_i the erasure rate and $i = 1, ..., L$ and E_i, $i = 1, ..., L$ is the number of redundant packets sent on that link. Let $Z_i^{(ns)}$ and $Z_i^{(s)}$ denote the number of original packets decoded upon completion of $K+E_i$ transmissions in the non systematic case (ns) and systematic case (s), respectively. We let U denote the number of received uncoded packets, $U \sim bin(K, p_i)$, and C the number of received random linear combinations, $C \sim bin(E_i, p_i)$. Then, the random network coding average residual erasure rates for the two cases at the i-th hop are given as

$$\bar{\eta}_{i,RND}^{(ns)} = 1 - Pr(Z_i^{(ns)} = K). \quad (4)$$

$$\bar{\eta}_{i,RND}^{(s)} = 1 - \frac{1}{K}\sum_{z=1}^{K} zPr(Z_i^{(s)} = z). \quad (5)$$

with

$$Pr(Z_i^{(ns)} = K) = \sum_{n=K}^{K+E_i}\prod_{j=0}^{K-1}(1-q^{j-n})Pr(N_i = n), \quad (6)$$

where N_i is a binomial random variable distributed as $N_i \sim bin(K+E_i, p_i)$ and

$$\begin{aligned} &Pr(Z_i^{(s)} = x) = \\ &Pr(U=x)\Big(1 - \sum_{c=K-x}^{E_i} Pr(C=c)\prod_{j=0}^{K-x-1}(1-q^{j-c})\Big) \\ &Pr(Z_i^{(s)} = K) = \\ &\sum_{x=0}^{K} Pr(U=x)\sum_{c=K-x}^{E_i} Pr(C=c)\prod_{j=0}^{K-x-1}(1-q^{j-c}), \end{aligned} \quad (7)$$

therefore

$$\bar{\eta}_{RND}^{(ns)} = 1 - \prod_{i=1}^{L}\Big(1 - \bar{\eta}_{i,RND}^{(ns)}\Big), \quad (8)$$

$$\bar{\eta}_{RND}^{(s)} = 1 - \prod_{i=1}^{L}\Big(1 - \bar{\eta}_{i,RND}^{(s)}\Big). \quad (9)$$

B. Codes with structure based on Pascal matrices

Our functional design can instantiate network codes with Pascal matrices in systematic and non-systematic mode with either deterministically or randomly chosen columns and different packet forwarding policies. In [10] we have derived the analytical expressions for Pascal network codes in systematic mode with deterministically chosen columns as

$$\bar{\eta}_{i,PAS}^{(ps)} = 1 - \frac{1}{K}\sum_{z=1}^{K} zPr(Z_i^{(ps)} = z), \quad (10)$$

$$\bar{\eta}_{i,PAS}^{(pc)} = 1 - \frac{1}{K}\sum_{z=1}^{K} zPr(Z_i^{(pc)} = z), \quad (11)$$

where we let $Z_i^{(ps)}$ and $Z_i^{(pc)}$ denote the number of original packets decoded in the Pascal systematic case (ps) and systematic case with scheduling (pc), respectively. While in the systematic case, coded packets follow the uncoded packets, scheduling of packets allows to control the delay. The following expressions are different arrangements of the derivations in [10], which assume two sub-blocks for scheduling illustrated in Fig. 7. Let K_s and n_{c_s} denote the number of uncoded and coded packets of the s-th sub-block $(s = 1, 2)$ transmitted by the encoder. We denote $U_s \sim bin(K_s, p_i)$ and $C_s \sim bin(n_{c_s}, p_i)$ as the random number of uncoded and coded packets of the s-th sub-block received, respectively, and $Z_{i_1}^{(pc)}$ as the random number of original packets of the first sub-block decoded at the node i.

$$Pr(Z_i^{(ps)} = x < K) = Pr(U=x)\Big(1 - \sum_{c=K-x}^{E_i} Pr(C=c)\Big),$$

$$Pr(Z_i^{(ps)} = K) = \sum_{x=0}^{K} Pr(U=x)\sum_{c=K-x}^{E_i} Pr(C=c),$$

with $U \sim bin(K, p_i)$ and $C \sim bin(E_i, p_i)$ the number of uncoded and coded packets received at the node i, respectively, and

$$\begin{aligned} &Pr(Z_i^{(pc)} = x < K_1) = \\ &\quad \sum_{u_1=0}^{x} Pr(Z_{i_1}^{(pc)} = u_1)Pr(Z_i^{(pc)} = x | Z_{i_1}^{(pc)} = u_1), \end{aligned}$$

$$\begin{aligned} &Pr(Z_i^{(pc)} = K_1 \le x < K) = \\ &\quad \sum_{u_1=0}^{K_1-1} Pr(Z_{i_1}^{(pc)} = u_1)Pr(Z_i^{(pc)} = x | Z_{i_1}^{(pc)} = u_1) \\ &\quad + Pr(Z_{i_1}^{(pc)} = K_1)Pr(Z_i^{(pc)} = x | Z_{i_1}^{(pc)} = K_1), \end{aligned}$$

$$\begin{aligned} &Pr(Z_i^{(pc)} = K) = \\ &\quad \sum_{u_1=0}^{K_1-1} Pr(Z_{i_1}^{(pc)} = u_1)Pr(Z_i^{(pc)} = K | Z_{i_1}^{(pc)} = u_1) \\ &\quad + Pr(Z_{i_1}^{(pc)} = K_1)Pr(Z_i^{(pc)} = K | Z_{i_1}^{(pc)} = K_1), \end{aligned}$$

with

$$Pr(Z_{i_1}^{(pc)} = u_1 < K_1) = Pr(U_1 = u_1)\Big(1 - \sum_{c_1=K_1-u_1}^{n_{c_1}} Pr(C_1 = c_1)\Big),$$

$$\begin{aligned} &Pr(Z_i^{(pc)} = x < K | Z_{i_1}^{(pc)} = u_1 < K_1) = Pr(U_2 = x - u_1) \\ &\quad \times\Big(1 - \sum_{c_1=0}^{K_1-u_1-1} Pr(C_1 = c_1)\sum_{c_2=K-c_1-x}^{n_{c_2}} Pr(C_2 = c_2)\Big) \end{aligned}$$

$$Pr(Z_{i_1}^{(pc)} = K_1) = \sum_{u_1=0}^{K_1} Pr(U_1 = u_1)\sum_{c_1=K_1-u_1}^{n_{c_1}} Pr(C_1 = c_1),$$

$$\begin{aligned} &Pr(Z_i^{(pc)} = x < K | Z_{i_1}^{(pc)} = K_1) = \\ &\quad Pr(U_2 = x - K_1)\Big(1 - \sum_{c_2=K-x}^{n_{c_2}} Pr(C_2 = c_2)\Big), \end{aligned}$$

$$\begin{aligned} &Pr(Z_i^{(pc)} = K | Z_{i_1}^{(pc)} = u_1 < K_1) = \\ &\sum_{u_2=0}^{K_2} Pr(U_2 = u_2)\sum_{c_1=0}^{K_1-u_1-1} Pr(C_1 = c_1)\sum_{c_2=K-u_1-c_1-u_2}^{n_{c_2}} Pr(C_2 = c_2), \end{aligned}$$

$$Pr(Z_i^{(pc)} = K | Z_{i_1}^{(pc)} = K_1) = \sum_{u_2=0}^{K_2} Pr(U_2 = u_2)\sum_{c_2=K_2-u_2}^{n_{c_2}} Pr(C_2 = c_2),$$

therefore

$$\bar{\eta}_{i,PAS}^{(ps)} = 1 - \prod_{i=1}^{L} \left(1 - \bar{\eta}_{i,PAS}^{(ps)}\right), \qquad (12)$$

$$\bar{\eta}_{i,PAS}^{(pc)} = 1 - \prod_{i=1}^{L} \left(1 - \bar{\eta}_{i,PAS}^{(pc)}\right). \qquad (13)$$

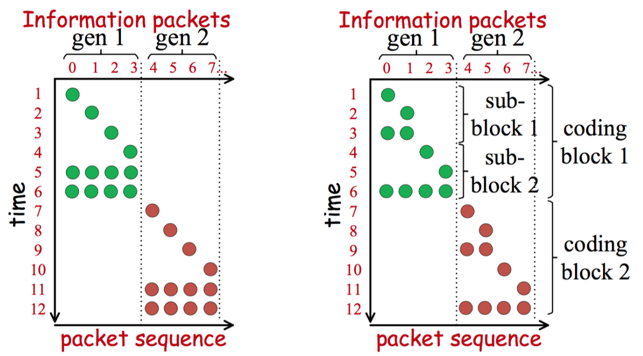


Fig. 7: Illustration of PascalNC and PascalNC-S for K = 4, N = 6. Each single dot represents an uncoded packet while multiple dots on a line represent a coded packet.

V. SatNetCode Testbed

We have implemented a line network with network coding applied intra-session over a video traffic input flow. Figs. 8 and 9 show the testbed setup for SatNetCode with one source node, three and five intermediate nodes and one destination node, respectively. It consists of four laptop computers, one fanless mini computer IPC3 and four single- board computers ODROID-XU4. The laptop computers are used for OBS, REL, INM and DES while the single board light weight ODROIDs are used only for network emulation. IPC3 and laptops run the same native C++ application implemented using the Qt cross-platform application framework.

All the ODROIDs (used for network emulation) run the Netem tool, which is included in Linux kernel to alter networking properties and emulate bandwidth, delay and packet losses. There is a network emulator controller (laptop computer) to configure Netem properties of all the Odroids used for network emulation. The specification of the testbed hardware is as follows

- **First Node (Laptop)**: CPU (Intel Core i7@2.6GHz), Memory (16G DDR4), Dimensions (18.9 x 28.3 x 1.7)
- **Second Node (IPC3)**: CPU (Intel Core i7@2.7 GHz), Memory (4G SDRAM), Dimensions - inches (7.4 x 6.2 x 1.5)
- **Third and Fifth nodes (Laptop)**: CPU (Intel Core i5@2.6 GHz), Memory (4G SDRAM), Dimensions - inches (14.9 x 9.9 x 1.2)
- **Fourth node (Lightweight Laptop)**: CPU (Intel Core i3@1.3 GHz), Memory (2G DDR3), Dimensions (11.4 x 8.2 x 0.93)

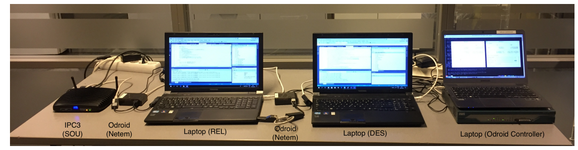


Fig. 8: SatNetCode testbed with three nodes (one intermediate node).

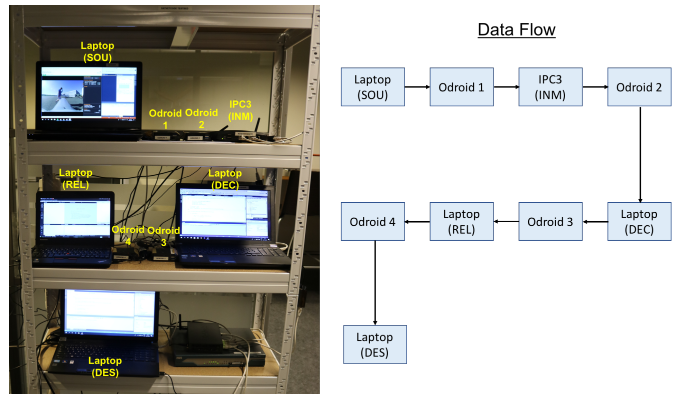


Fig. 9: SatNetCode testbed with five nodes (three intermediate nodes).

VI. Experimental results

In this section we present experimental performance analysis of the network codes with our implementation of C-NC functions.

A. Structural and operational properties

First of all, we run a two-hop symmetric scenario in order to analyze the structural and operational properties of the network codes. A realistic scenario with two hops is shown in Fig. 1. We assume realistic bandwidth limitation in the first (SATCOM) hop, which we model with a limited bandwidth of 250 kbps. We assume coded payload and metadata encapsulated in the payload of an IP packet. Links are assumed with uncorrelated 5% erasure rates. Overall system/network latency is set to 500 msec (assuming worst propagation case of GEO orbit). Note that overall delay has three contributions: system/network latency due to propagation and lower layer protocol dynamics, and coding algorithmic latency, which mainly depends on the implementation efficiency and the network code scheme. Here we show the tests for the following basic network coding schemes: 1) regular network coding scheme **SNC** with repair packets sent after systematic packets, 2) scheme with random coefficients and packet scheduling at the intermediate node as described in [8], which we denote as **FuncNC-Random** and 3) scheme using Pascal matrices with no packet scheduling, which we denote as **FuncNC-Pascal** and that can be in non-systematic (Non-Sys) and systematic mode (Sys).

1) Analysis of Structural Properties: We have implemented the systematic and non-systematic versions of Pascal. In our systematic instantiation, the intermediate node probes received (uncoded and repair) packets. If all the K packets are recovered then the encoder sends uncoded and repair packets to the next node or else only the recovered systematic packets are passed to the packet scheduler. Fig. 10 shows the structural properties for $R = N/K$ as a function of the coding window and Fig. 11 as a function of $R = N/K$. We can conclude from the results that from a structural point of view Pascal codes show as good as or better performance than the codes with random structure.

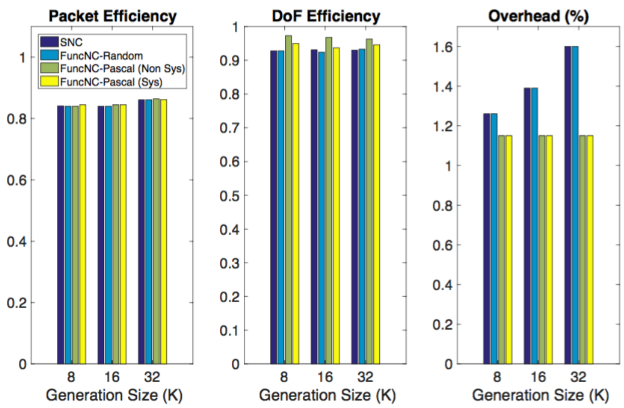


Fig. 10: Structural properties for $R = 1.2$ as a function of coding window size for $\epsilon_1 = \epsilon_2 = 5\%$.

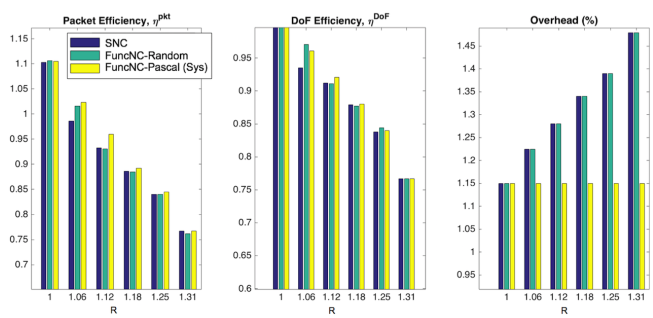


Fig. 11: Structural properties for $K = 16$ as a function of R for $\epsilon_1 = \epsilon_2 = 5\%$.

2) Analysis of Operational Properties: First, we show operational gains of re-encoding in Fig. 12. This figure shows the optimal redundancy required by FuncNC-Random with and without re-encoding to achieve a target residual rate of $\eta^{res} = 1\%$. The results are shown for the two symmetric cases $\epsilon_1 = \epsilon_2 = 5\%$ and $\epsilon_1 = \epsilon_2 = 10\%$.

The results show that with network re-encoding, we can achieve up to 43.9% gain. Hence, re-encoding not only brings higher achievable throughput, but also smaller traffic overhead. The gain is expected to be higher for higher number of intermediate nodes.

Fig. 13 shows the estimated probability and cumulative density functions over 10000 packets. It can be observed the different behaviour of non systematic Pascal, which shows a Gaussian-like probability density due to the fact that the intermediate node decodes in order to forward all decoded packets. This is different to usual systematic operation, where packets are progressively probed, decoded and forwarded. Accordingly, we observe the probability peak delay of systematic

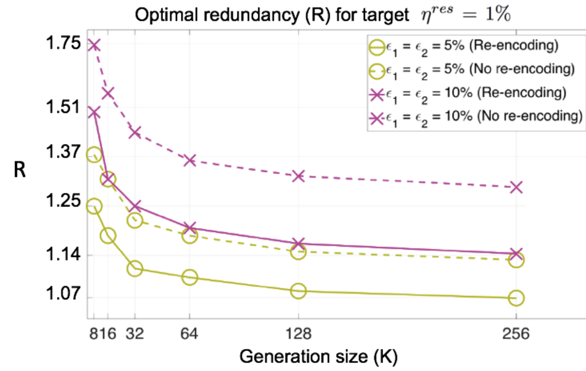


Fig. 12: Optimal R for FuncNC-Random with different generation sizes for two erasure channel cases $\epsilon_1 = \epsilon_2 = 5\%$ and $\epsilon_1 = \epsilon_2 = 10\%$.

packets, followed by the smaller peaks of those lost packets that need to wait for repair packets to arrive to be successfully decoded. As expected, non-systematic Pascal shows larger delays, pointing to the reliability-delay trade-off as we will see later. Note that average values are below theoretical upper bounds shown in Fig. 2.

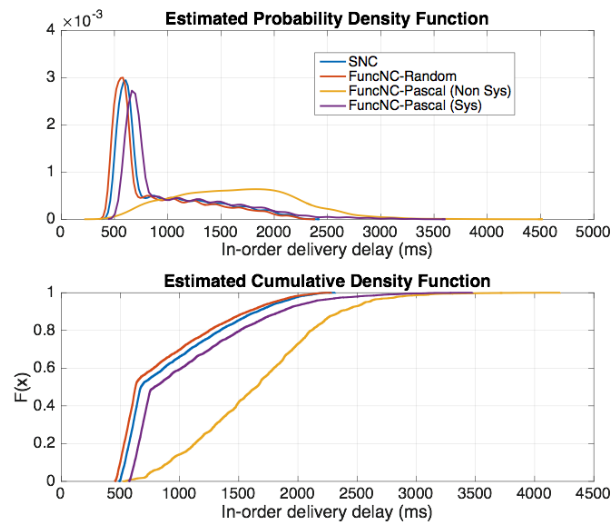


Fig. 13: Estimated Probability Density Function and Cumulative Density Function for $\epsilon_1 = \epsilon_2 = 5\%$ and $R = 1.2$.

Fig. 14 shows some numerical values of residual erasure rate. We observe that as expected the non-systematic Pascal has the worst performance at low coding window sizes. However, at the convenient window size of 16, the best performing code is systematic Pascal, which points out to the benefit of structured codes even in SATCOM scenarios.

Note that the FuncNC-Random version under testing has worse performance than SNC in residual rate but has better delay performance and lower complexity [8], which again points out the delay-reliability trade-off.

B. Experimental results for realistic scenarios

We carried out multiple tests assuming realistic scenarios, show in Fig. 1 and Fig. 16 for two and three hops, respectively.

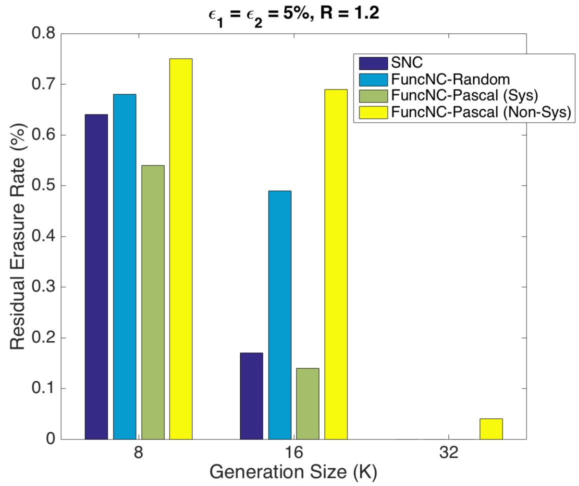


Fig. 14: Residual Erasure Rate for $\epsilon_1 = \epsilon_2 = 5\%$ and $R = 1.2$.

In both cases we assume that video data is captured by the drone or the mobile device. This node and network coding is performed over the video data. The UAV can support communication using mobile SatCom terminal (for e.g., using COBHAM AVIATOR UAV200). The UAV sends data to the decision node at control center over the satellite link. The decision node provides a visualization of the received observations and also forward the relevant observations to the end node. The node at the decision facilities includes both network decoding and network re-encoding functionalities. This node can transmit packets using both cellular or SatCom networks depending on the availability of the networks for the field users. Finally, the video data is received by the action node. The action node includes both network decoding and video decoding functionalities. The scenarios are emulated in the SatNetCode test-bed using the network parameters specified in Table I.

1) Optimal achievable rates: However, during the execution of our project, we realised that the designer does not know which coding rate to use to achieve target residual erasure rates in an efficient way. Therefore, it is imperative for the designer of network coded services to know what are the theoretically optimal achievable rates for a given target residual erasure rate. We note that this information was lacking in the literature since most of the works focus on theoretical rates using very long (capacity achieving) codes, which is not realistic due to the fact that for practical implementation, network codes have finite-length. Therefore, it is very important in order to reduce unnecessary complexity or delay using bigger generations than the strictly needed. We also note that such an optimization is not a trivial task due to in-network re-encoding and dynamic topology. We have developed in [10] novel optimization procedures in [10] and [11], which are examples of optimization depending on the service aim that network coding is used for. Here, we consider for validation the following optimization from [10].

Given N and a target residual erasure rate denoted as P_e^0, the optimal coding rate denoted as ρ^*, is defined as

$$\begin{aligned} \rho^* &= \max_{\rho \in \Psi} \rho \\ \text{s.t.} \quad & P_e^L(\mathcal{D}, \rho) \le P_e^0, \end{aligned} \tag{14}$$

where Ψ is the set of coding rates available for searching, $\rho[i] = i\rho_0$, $1 \le i \le |\Psi|$ (e.g., $\rho_0 = \frac{1}{N}$, $\rho[|\Psi|] = \frac{N-1}{N}$). $P_e^L(\mathcal{D}, \rho)$ is the PLR at the destination according to each network code. For L hops, we denote the optimal achievable rate as

$$R^* = \rho\big(1 - P_e^L(\mathcal{D}, \rho)\big). \tag{15}$$

Theoretical values are illustrated in Fig. 16, where it is also shown the (mincut) capacity of the topology. We have chosen the following network codes for this theoretical study:

- **SNC and SNC-S**. For SNC, the tested implementation sends systematic packets first and then the coded packets while coded packets (the linear combination of systematic packets using random coefficients) are scheduled every sub-block of packets in SNC-S. In our implementation, that the random coefficients are sent along the packets.
- **PascalNC and PascalNC-S**. For PascalNC, the tested implementation sends systematic packets first and then the coded packets while coded packets (the structured combination of systematic packets using Pascal columns) are scheduled every sub-block of packets in PascalNC-S. In our implementation, we assume offline optimization and therefore zero overhead.

We observe that:

- The optimal coding rate increases exponentially with $N \le 100$. The optimal coding rate has an exponential behavior and for this reason tightness in the theoretical results is difficult to achieve.
- As expected, the achievable rate is closer to the capacity for higher block sizes. Also as expected, the higher rates towards achievability for large block lengths correspond to the constructions with no scheduling since the effective coding length is smaller.
- The optimal coding rate from Pascal network coding is higher than the other schemes due to the effect of decoding and encoding at intermediate nodes.
- Due to the size of the Pascal matrix (limited to 256 for the test-bed), Pascal network coding needs to operate with prescribed blocklength. In contrast, SNC can work with any block length depending on the design constraints such as pakcket delay and computational resources e.g., hardware capability, energy consumption. However, for our test-bed and target use-cases, the size of 256 is more than sufficient since we will need generation sizes or the order $N \le 64$.

Fig. 18 shows one example of the validation we carried out of the theoretical optimal coding rates. We have chosen the validation the following network codes:

- **FuncNC-random network codes**, which correspond to the SNC described above.
- **FuncNC-Pascal network codes**, which correspond to the PascalNC described above.

We observe that

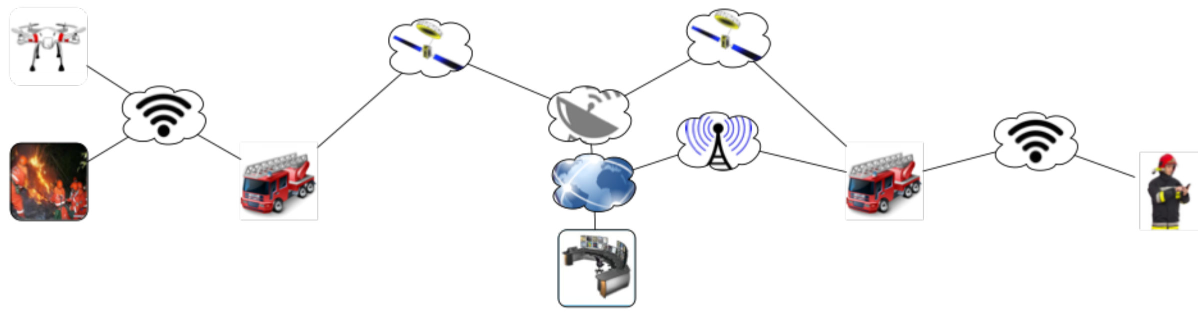
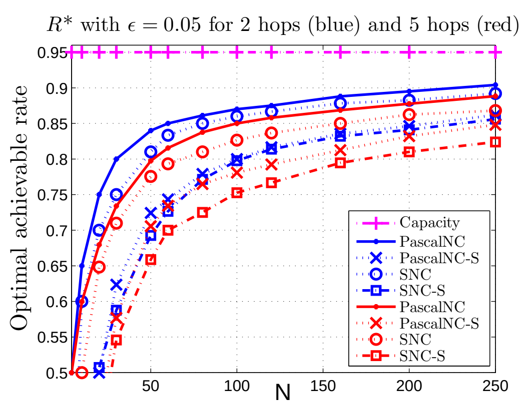


Fig. 15: Test-Case Scenario for 5 nodes. First and the last links are WiFi networks. Whereas second and third links correspond to the Satellite links.

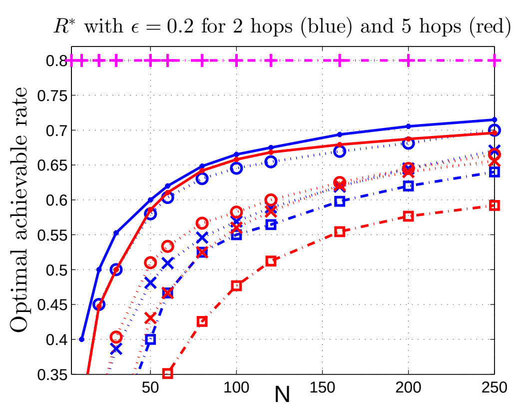
	Network Type	Bit Rate	Latency	Packet Erasure Rate
First Link	WiFi	10 Mbps	20 msec	5% or 10%
Second Link	SatCom	200 Kbps	500 msec	5% or 10%
Third Link	SatCom	200 Kbps	500 msec	5% or 10%
Fourth Link	WiFi	10 Mbps	20 msec	5% or 10%

	Network Type	Bit Rate	Latency	Packet Erasure Rate
First Link	SatCom	200 Kbps	500 msec	5% or 10%
Second Link	Cellular	1 Mbps	50 msec	5% or 10%

TABLE I: Testbed network parameters for the realistic use-case scenarios with three (top) and five (bottom) nodes.



(a)



(b)

Fig. 16: Finite-length theoretical optimal achievable rate at the destination node for two different erasure rates and $P_e^0 = 10^{-6}$.

Generation Size (K)	Target Residual Erasure Rate	Optimal Coding Rate (Func-NC Random network code)	Optimal Coding Rate (Func-NC Pascal network code)
8	10^{-2}	0.7272	0.8
16	10^{-2}	0.8	0.8421
32	10^{-2}	0.8648	0.8888
64	10^{-2}	0.8888	0.9014

TABLE II: Theoretical optimal coding rates.

2) Experimental rates: First, to validate our theoretical results, we have derived optimal network coding rates using equations (14) and (15) with target residual erasure rate of 10^{-2} and per-link erasure rate of 5% for different generation sizes $K \in \{8, 16, 32, 64\}$. The theoretical results are shown in Table II.

Using the same network coding parameters (generation sizes and optimal coding rates) and with the same setup (5% per-link erasure rate), we have performed 10 iterations per experiment to calculate then residual erasure rate and then compared the experimental results with the theoretical target residual erasure rate of 10^{-2} as shown in Fig. 17.

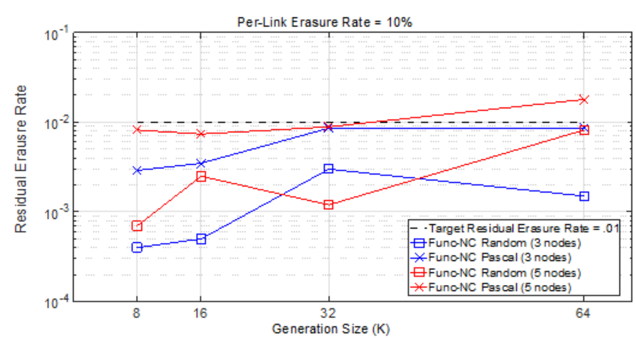


Fig. 17: Residual erasure rate for FuncNC-Pascal and FuncNC-random schemes for 3-nodes (blue) and 5-nodes (red) line network with per-link erasure rate of 10% and target residual erasure rate of 1%.

Our results show that in almost all the cases, experimental residual erasure rate is smaller than the theoretical target residual erasure rate and hence the objective of meeting the target residual erasure rate is achieved by using the optimal coding rates that are derived using our theoretical framework. The optimal coding rates from theory are conservative as the experimental residual erasure rates are met from below however since the optimal coding rate has an exponential behavior (Fig. 16), the tightness in the results is difficult to achieve.

Second, using the same network coding parameters (Table II), we also perform experiments to compare the in-order average delivery delay for FuncNC-Pascal and FuncNC-random network coding schemes. Fig. 18 presents the experimental delay values from both the coding schemes. One of the major components of the overall delay is the network delay which is due to the accumulation of latency of different networks in the line network topology. For example, in case of 3-nodes, network delay is 550 msec (first link has SatCom network with latency 500 msec and second link has Cellular network with latency 50 msec as shown in Table I) and in case of 5-nodes, network delay is 1040 msec (first and fourth links have WiFi network with 20 msec latency each and second and third links have SatCom network with 500 msec each as shown in Table I).

Our results show that the in-order average delivery delay is higher in FuncNC-Pascal network coding scheme as compared to FuncNC-random scheme. This is because FuncNC-Pascal scheme requires additional complexity and decoding time at each intermediate node while FuncNC-random scheme does not require decoding at intermediate nodes. Our results show the rate-delay tradeoff of using FuncNC-Pascal network coding scheme, where, it provides higher rates but also higher delay as compared to FuncNC-Random network coding scheme. However, the increase in delay is minimal for small generation sizes where for $K = 8$, the delay from Pascal-NC is 638 msec and the delay from Pascal-Random is 634 msec for 3-nodes setup, which is only 0.63% increase. Similarly, the increase is only 1.85% when $K = 16$. Since, the major component of the overall delay is the network delay, both coding schemes give similar performance for small generation sizes.

Fig. 18: In-order average delivery delay for FuncNC-Pascal and FuncNC-random schemes for 3-nodes (blue) and 5-nodes (red) line network with per-link erasure rate of 5% and target residual erasure rate of 1%.

VII. Conclusions and Further Work

We have presented a per-node functional architecture and experimental testbed for the design and validation of network coding over satellite. The design allows to instantiate different coding schemes with network coding coefficients locally or remotely configured.

We have presented our experimental testbed validation. Our results show the practical benefits of re-encoding and performance tradeoffs of network coding schemes with random and with fully structured coding coefficients. In particular, our results show the benefits of structured codes, which show good structural properties and good delay-reliability trade-off due to the regenerative properties of coding coefficients.

Our first tests presented here have been preformed for a worst case of bandwidth-limited GEO scenario. Even for such a conservative case, we have proved the high relevance of using network coding for increasing reliability with good delay-reliability trade-off using random or structured codes. Hence, our results are expected to be far more promising for optimized code instantiations and other SATCOM scenarios (e.g. lower orbits). Further work includes a comprehensive experimental tests of the complete functional architecture with multicast and video traffic.

Acknowledgment

This work has been supported by the European Space Agency, under the ESA Contract number 4000115046/15/LS/US: SatNetCode, and we acknowledge the excellent support and decisions with the ESA project officer Dr. Ilias Panagiotopoulos.

References

[1] R. Ahlswede, N. Cai, S. -Y.R. Li , R. W. Yeung "Network information flow" *IEEE Trans. on Inf. Theory*, vol. 46, no. 4, pp 1204-1216. Jul. 2000
[2] *International Journal of Satellite Communications and Networking*, Special Issue: Network Coding Applications in Satellite Networks. vol. 35, no. 6, pp. 533-620, November/December 2017.
[3] J. Cloud and M. medard, "Network coding over SATCOM: lessons learned," WiSATS 2015.
[4] https://artes.esa.int/projects/satnetcode
[5] T. K. Dikaliotis, A. G. Dimakis, T. Ho, M. Effros, "On the delay of network coding over line networks", ISIT 2009.
[6] A. Vazquez-Castro, L. M. Contreras-Murillo, "Softwarization of Network Coding Functions and Logical Mapping to SDN", ISNCC 2018.
[7] B. Shrader, J. Brooke, M. Nathaniel, "Systematic wireless network coding", MILCOM 2009.
[8] P. Saxena and M. A. Vazquez-Castro, "DARE: DoF-Aided Random Encoding for Network Coding over Lossy Line Networks", *IEEE Communications Letters*, vol. 19, no. 8, pp. 1374-1377, Aug. 2015.
[9] S. Yang and R. Yeung, "Coding for a network coded fountain," ISIT 2011.
[10] Tan Do Duy, A. Vazquez-Castro, "Finite-length performance comparison of network codes using random vs Pascal matrices", Submitted to IEEE Wireless Communications Letters, 2018.
[11] Tan Do Duy, A. Vazquez-Castro, "Geo-controlled Network Coded Video Streaming over Converged Satellite-Cloud Networks", under review in IEEE Transactions on Aerospace and Electronic Systems, 2018.
[12] M. Hua, S. B. Damelin, J. Sun, M. Yu, "The Truncated & Supplemented Pascal Matrix and Applications", *Involve Journal of Mathematics*, vol. 11, no. 2, 243-251. 2018.
[13] S. Hemminger, "Network Emulation with Netem," Proceedings of the Linux Conference Australia (LCA), April 2005