Joint Control of Delay and Packet Drop Rate in Satellite Systems using Network Coding

Sameh Sorour*, Shahrokh Valaee* and Nader Alagha†

*Department of Electrical and Computer Engineering, University of Toronto, Canada
Email:{samehsorour, valaee}@comm.utoronto.ca

†European Space Agency, Netherlands
Email:nader.alagha@esa.int

***Abstract*—Mobile terminals communicating through satellite suffer from low channel quality due to the combination of slow and fast fading and limited battery power. Moreover, the advanced applications proliferating over such terminals forbids the use of feedback based packet retransmission schemes over satellite due to their high latency, exceeding three times the round trip time. These feedback based packet retransmission schemes also suffer from large throughput degradations to reduce packet drop rates, especially for the large receiver population.**

In this paper, we propose a joint delay and packet drop rate control protocol over lossy mobile satellite channels using network coding. The suggested protocol employs random network coding at the mobile terminals, within and across different sessions, to generate efficient proactive retransmission packets, without prior knowledge of lost packets at the different users. It also allows the satellite to transmit random network coding combinations of all received packets, to fill the gaps left by packet losses on the uplink channel. By adjusting the timing of these network coded transmissions, the protocol can control packet recovery at any desired delay above one round trip time. The protocol can also control the packet drop rate level by adjusting the network coding rate. The protocol is able to achieve these gains at a much higher system throughput compared to conventional ARQ protocols. Moreover, the protocol do not suffer from the lack of throughput scalability for larger user populations.

***Index Terms*—Satellite Systems; Network Coding; Delay Control; Packet Drop Rate Control.**

I. Introduction and Motivation

The immense advances in communication technology and its deep penetration in human activities made the wireless connectivity anywhere, anytime to anybody nonnegotiable demand. Pedestrians, vehicle, train and even flight passengers want to be connected to each other anytime. Great advances in cellular networks helped in carrying some of this massive demand. However, cellular networks may not be enough to carry this tremorous data burden, especially for applications requiring fast and reliable communications between mobile users spread over geographically large areas. This urges the need to employ geostationary (GEO) satellite systems as potential means to provide such services.

Basic mobile communication services through satellite systems has emerged in the past two decades. However, current advances call for a new level of high speed, multimedia and interactive applications, in which mobile terminals could communicate multiple high speed sessions with different subsets of other mobile terminals. Some of these applications require reliable delivery of packets with a target packet drop rate, and before a certain deadline. Examples of such applications are video conferences, online gaming, in which parties may be located at different continents. In these interactive applications, it is very important to guarantee a minimum level of packet reception with low delay, in order to achieve full parties' awareness of their peers' actions. Moreover, the proliferating high-definition peer-to-peer video streaming applications through satellite impose such delay and packet drop rate (PDR) constraints.

One major challenge against obtaining high speed and reliable communications among mobile terminals through satellite is the very poor wireless channel. Unlike conventional satellite gateways, mobile devices are subject to more aggressive fading and channel fluctuations due to mobility, shadowing and multipath reception. Moreover, they do not have the high power capabilities found in satellite gateways since they are powered by low life batteries. All these effects render the channel qualities very poor on both uplink and downlink, and thus packets will incur high loss rates on both uplink and downlink satellite channels. Consequently, applications with PDR constraints require the use of strong and controllable packet recovery schemes over satellite channels.

One solution is the use of Go-Back-N or selective-repeat automatic repeat request protocols (GBN-ARQ, SR-ARQ, respectively) with accumulated, selective or negative packet acknowledgements, over a satellite reverse link [1]–[3]. In both protocols, the packet recovery phase starts only when the satellite or the mobile terminals determine whether they correctly detected the packets or not. This results in a full round trip time (RTT) delay between a packet transmission and its retransmission, leading to three RTT overall delay. This creates another problem for the applications of interest, as they mandate low packet delays.

Several works have proposed different solutions for these problems [4]–[8]. The first two works proposed a peer-to-peer inter-receiver retransmission scheme.However, this solution necessitates the geographical collocation of the mobile terminals, and adds complexity to the mobile terminals for clustering and exchanging information. [6] suggested the implementation of ARQ over terrestrial wired or wireless links. This solution is quite inadequate for mobile terminals, as it will greatly overload the terrestrial wireless networks with packet retransmission, especially in scenarios with large number of mobile terminals. [7], [8] proposed a modification of the previous solution, in which the terrestrial reverse links are employed only for packet acknowledgments.. For this type of architecture, we proposed an network-coded ARQ (NC-ARQ) protocol in [9], that employs both proactive and reactive retransmissions using linear network coding. Despite its numerous gains above SR-ARQ, in terms of ordered packet delay, packet drop rate and average throughput, the proposed protocol does not provide PDR and delay guarantees.

The use of proactive retransmissions, without prior knowledge of the lost packets in satellite systems, was proposed in [10]. This work proposed the use of packet level forward error correction (FEC) using block coding. Another similar solution is the use of digital fountain codes [11], [12]. However, these works did not present structured retransmission protocols for the applications of interest in satellite systems. Moreover, these techniques are generally known to be applied to packets belonging to the same session and destining to the same subset of terminals. Also, none of these approaches extend readily to continuous decoding-less relaying through lossy two-link tandem channels found in satellite systems. On one hand, the satellite should wait to receive the whole FEC encoded block at the sender to either decode and re-encode, or concatenate another FEC code on it. This slows down the downlink stream and incurs additional non-desirable delay. On the other hand, if the satellite applies a fountain code on the fountain coded packets it has received so far, this does not mean that the overall code, from the sender to the mobile terminals, will have the properties of a fountain code [13]. In other words, fountain codes are not composable across links. A decode and re-encode protocol will be sub-optimal in terms of delay, as pointed out by [14].

Recently, network coding has emerged as a routing and scheduling scheme that attains maximum information flow in a network. The main core of network coding is the idea of packet mixing among the packet of the same and/or different data flows, using several techniques such as packet XOR [15], [16] and linear coding [17]. One of the major gains of network coding arise from the concept of overhearing other users' packets in order to employ them in decoding more efficient inter-flow packet combinations. Despite all its benefits, network coding has not been studied in the context of satellite networks.

In this paper, we propose a *network coding protocol for proactive retransmissions in satellite systems, to jointly control delay and PDR*. The suggested protocol employs random network coding at different terminals, to generate efficient proactive retransmission packets within and across different sessions, without prior knowledge of lost packets at the satellite and the destinations. Since all destination terminals can overhear all the packets broadcasted by the satellite, they can efficiently decode the packets of all mixed sessions and then extract their own packets latter. At the satellite, our protocol allows two mode, namely the transparent and regenerative modes. In the former mode, the satellite just amplifies each received packet from the sending terminal and broadcasts it to all terminals in the network. In the latter mode, if a packet is lost on the uplink channel, the satellite encodes all previously received packets, using random network coding, then transmits the resulting packet instead of this lost packet in the uplink. Unlike fountain codes, random network coding is composable [18], thus allowing the satellite to apply packet combinations on previously combined packets without decoding them. By adjusting the timing of the network coded transmissions generated at the sending terminal, the protocol can control packet recovery at any desired delay above one system RTT. The protocol can also control the PDR level by adjusting the network coding rate. Our simulation results show that a PDR of as low as 10^{-6} can be achieved with a considerably high throughput compared to conventional ARQ schemes, when the individual packet loss rates at the uplink and downlink channels are as high as 10^{-1}. These gains come at the cost of decoding complexity, decoding delay and coding coefficients' overhead. Nonetheless, the maintenance of global delay constraints reduces the decoding delay drawback of our protocol. Moreover, advanced digital signal processors can greatly reduce the decoding complexity of random network coding. Finally, the new encapsulation modes in satellite systems can easily embrace the network coding coefficient overhead with minor changes.

The rest of the paper is organized as follows. In Section II, the satellite system model and parameters are introduced. Our proposed network coding protocol, for joint control of delay and PDR, is introduced in Section III. Simulation results comparing our proposed protocol to SR-ARQ and NC-ARQ are depicted in Section IV. Section V concludes the paper.

II. System Model and Parameters

In this paper, we consider a satellite network that consists of a GEO satellite and multiple mobile terminals, which could be carried by a pedestrian, or a user in a car a train or even a plane, and thus are generally separated by large geographical distances. Each mobile terminal is having different communication sessions with different subsets of the other terminals. To organize the communication sessions of all terminals, the satellite assigns different timeslots to each of the terminal in a demand assigned multiple access (DAMA) scheme. During its assigned timeslot, the mobile terminal transmits to the satellite a group of its packets that belong to one or several sessions. The satellite role is then to deliver these packets to their destinations.

We call the channels from terminals to satellite and from satellite to terminals as the uplink and downlink channels, respectively. Let R_b be the channel bit rate for both uplink and downlink channels, L be the length of a packet, and $T_P = L/R_b$ be the packet transmission duration. We assume that the satellite and the different mobile terminals experience different physical shadowing and fading levels, which may corrupt the transmitted packets. From the data link layer perspective, this translates into packet loss events. We assume packets are lost in both uplink and downlink from and to the i-th mobile terminal with rates P_i^u and P_i^d, respectively. We define the round trip time (RTT) of the satellite system as the time from the start of a packet transmission from the sending terminal until the start of its reception at the destination terminals. In GEO satellites, the RTT is almost equal to 250 ms for all mobile terminals.

For ARQ based retransmission schemes, both the satellites and the mobile terminals report only negative acknowledgments. These will then be responded to by a packet retransmission, either un-coded as in SR-ARQ, or coded as in NC-ARQ.

In general, most of the delay constrained applications (such video streaming and interactive services) cannot usually employ received packets out of order. Consequently, packets following a missing packet are stored at the data link layer, and are not passed to upper layers, until the missing packet is correctly received or ignored due to its deadline expiry. Consequently, we define the experienced delay by correctly received packet as the interval between their transmission instant from the sender's data link layer until their ordered delivery to the upper layers at the mobile terminals.

III. Proposed Network Coding Protocol

In this section, we introduce our network coding protocol for proactive packet retransmission in satellite systems. The protocol design aims to jointly control the average delay of received packets and the PDR.

A. Frame Structure and Delay Control

In our proposed protocol, the temporal scale is partitioned into blocks, each including a fixed number $N = U + K$ of packets that can fit into one timeslot of duration T_B. The value of N depends on the block duration T_B, the packet length L and the employed modulation and coding scheme (and thus the channel bit rate R_b). Figures 1 and 2 depict the frame structure for the transparent and regenerative satellite modes, respectively.

At each terminal, data flows from different sessions are partitioned into segments, which are encapsulated and appended with cyclic redundancy check (CRC) to form source packets. Once a terminal is granted a timeslot, it transmits U packets $\{S_1, \ldots, S_U\}$ towards the satellite. These packets are generally destined to different subsets of the other terminals. In previous approaches, each session packets may be encoded together as they will be sent to the same destinations. In this case, each destination detects its own packets and decode them. However, all terminals are generally able to overhear all the packets sent by the satellite due to the broadcast nature of the downlink channel. In our protocol, we exploit this overhearing property to allow more efficient inter-session encoding using network coding. In other words, after the transmission of the

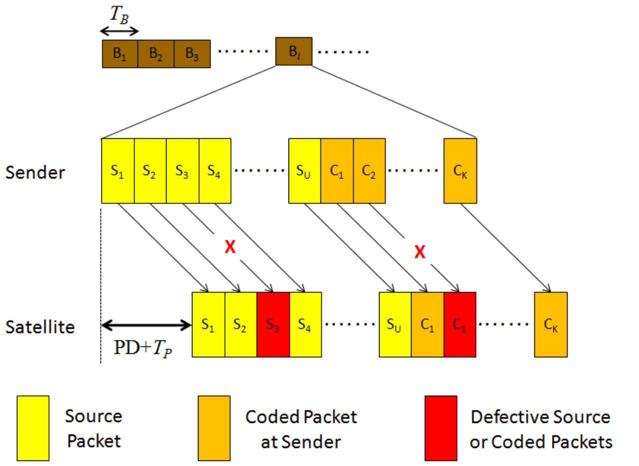

Fig. 1: Packet transmission diagram for transparent satellites

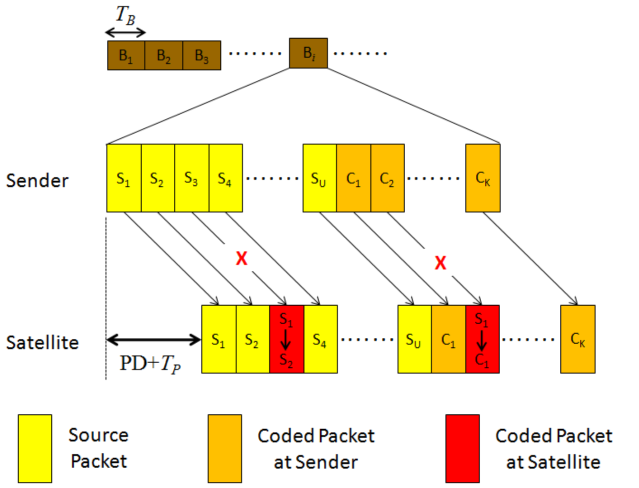

Fig. 2: Packet transmission diagram for regenerative satellites

U source packets, the sending terminal transmits K random network coded packets $\{C_1, \ldots, C_K\}$, generated from all the U source packets. All receiving terminals detect and decode all the N packets, then pass only their own intended packets to upper layers and discard the others. This is a famous type of network coding, which increases the throughput compared to sending separate packets for each subset of destinations, and has been proved to achieve network capacity in [18]. To avoid coding delay, the sending terminal can perform the coding operations sequentially over the payload of each generated packet, as will be explained in Section III-B, then the final encoded segment is encapsulated and appended with CRC to form the coded packet. Once the full block is transmitted, another terminal repeat the process in the following timeslot. Note that we employ a systematic network coding (SNC) scheme, as it achieves much better delay [19] and optimum PDR [20] compared to non-systematic network coding, for such rate based scenario.

At the satellite, the relaying process depends on the satellite mode. For the transparent mode depicted in Figure 1, the satellite has no processing capabilities other than amplify the received signal and broadcasting it on the downlink channel. Consequently, the satellite relays all received packets, whether they are received correctly or with errors at the satellite. Since the satellite requires to buffer the whole packet before broadcasting it, the time between the start of a packet transmission from the sender, and the start of its transmission from the satellite, is equal to the uplink propagation delay (PD = RTT/2) plus the packet transmission duration T_P. In the regenerative mode, the satellite possesses processing and coding capabilities. When a packet is received at the satellite, it first checks its correctness using the CRC bits. If the packet is correct, it is amplified and broadcasted on the downlink. If the packet is erroneous, the satellite transmits a random network coded packet instead, including a combination of all previously received packets of the block. The coding should be again done sequentially to avoid coding delay at the satellite. Once a full block is transmitted, the satellite discards its packets and proceeds to the next block.

From the above transmission scheme, the receiver will finish the reception (either correct or not) of all the packets transmitted by the satellite for one block after RTT + $T_B + T_P$ time units. Since RTT and T_P are mandated by the system delay and rate parameters, the value of T_B can be selected by the protocol designer, so as to guarantee the termination of a block reception, before the imposed deadline. In other words, delay control is achieved by adjusting T_B to a value that allows the full decoding of all packets, the filtering of own packets, and their passing to upper layers before the delay constraint. It will be explained in Section III-C that the number of coded packets K decreases (and thus the system throughput increases) when the coding is done over larger block sizes (i.e. larger N). Consequently, the value of T_B should be set in the protocol, so that the termination of a block reception is done at small duration before the deadline, just enough for packet decoding, filtering and passing to upper layers.

B. Packet Encoding and Decoding

In our protocol, we require retransmission packets to be sent without prior knowledge of the lost source packets. We can employ random network coding to generate these retransmission by combining all U source packets $[S_1, \ldots, S_U]$. Let $\alpha_i = [\alpha_{i,1}, \ldots, \alpha_{i,U}]$ be the coefficient vector for the i-th coded packet, whose elements are chosen from a Galois field of appropriate dimension. The i-th coded packet (C_i) can be generated as follows:

$$C_i = \sum_{j=1}^{U} \alpha_{i,j}\, S_j \ . \tag{1}$$

In general, the coefficient vectors employed to generate the different coded packets should be chosen such that any U packets, from the N packets of the block, should be linearly independent. We refer to this property as the decodability property. We assume that these coefficient vectors are generated from a large enough Galois field to guarantee the decodability property almost surely. This property is also more guaranteed in systematic network coding compared to non-systematic network coding, as the former employs less coefficient vectors than the latter, thus reducing their probability of linear dependence.

To speed up the coding process, the sending terminal can perform the K Gallois field multiplications $\alpha_{i,j} S_j \ \forall \ i \in \{1, \ldots, K\}$ once packet S_j is generated, then adds the results to their corresponding previously encoded packets. When this process is done for the U segments, the packets are directly generated by encapsulating the segments and appending CRC to them. This reduces the processing time for encoding compared to postponing it when all packets are generated.

It is necessary to include the coding vectors (or information needed to reconstruct them) in each coded packet header, in order for the mobile terminals to decode the packets. If the same random number generator is employed in all mobile terminals and the satellite (in case of regenerative mode), it is sufficient to include the seed that generated these coefficients, which requires only around four bytes per packet [21], [22]. Since satellite systems usually employ encapsulation protocols to carry network packets, these encapsulation protocols can be directly extended to support network coding [23]. Using the same approach as in [24] for the FEC support over Multiple Protocol Encapsulation (MPE), it possible to redefine certain header fields to carry network coding coefficients/seeds. In case of Generic Stream Encapsulation (GSE), the extension header mechanisms allow for the introduction of new fields to carry network coding coefficients/seeds. The use of existing encapsulation protocols to support network coding allows simplified deployment. Furthermore, the location of these protocols just below the network layer in the protocol stack allows network coding to be transparent to the higher layers including the network layer [23].

When a packet is lost on the uplink in the satellite regenerative mode, it encodes all previously received packets of the block and broadcasts them to the mobile terminals. Defining I_x as an indicator function, which is equal to 1 if packet x is correctly received at the satellite, and is equal to zero if it is not correctly or not yet received, the coding process at the satellite is performed using a new set of coefficient vectors $\beta_k = [\beta_{k,1}, \ldots, \beta_{k,N}]$ as follows:

$$\begin{aligned}
C_k^s &= \sum_{j=1}^{U} \beta_{k,j}\, S_j I_j + \sum_{i=1}^{K} \beta_{k,(U+i)}\, C_i I_{(U+i)} \\
&= \sum_{j=1}^{U} \beta_{k,j}\, S_j I_j + \sum_{i=1}^{K} \beta_{k,(U+i)} I_{(U+i)} \left(\sum_{j=1}^{U} \alpha_{i,j}\, S_j \right) \\
&= \sum_{j=1}^{U} \left(\beta_{k,j} I_j + \sum_{i=1}^{K} \alpha_{i,j} \beta_{k,(U+i)} I_{(U+i)} \right) S_j \\
&= \sum_{j=1}^{U} \psi_{k,j}\, S_j \ ,
\end{aligned} \tag{2}$$

The above equation shows that the application of a new network code over the existing one results in a new network code over the same set of source packets, with global coding coefficients $\psi_{k,j}$ equal to:

$$\psi_{k,j} = \beta_{k,j} I_j + \sum_{i=1}^{K} \alpha_{i,j} \beta_{k,(U+i)} I_{(U+i)} \ . \qquad (3)$$

After encoding, the satellite includes the global coefficient vector $\psi_k = [\psi_{k,1}, \ldots, \psi_{k,U}]$ inside the packet header and broadcasts it to the mobile terminals.

The satellite can always perform a sequential encoding process, as described above, to be prepared for any packet loss on the uplink. When the satellite receives coded packets from the sender, the global coefficient vectors should be directly updated with each added coded packet, to reflect the new value of the global coefficients. When a loss event occurs on the uplink channel, the satellite coded packet is ready for transmission without any delay. Since network coding is composable as shown in (2) and (3), the transmitted packet will have the innovation guarantee property. In other words, it will bring new information to every receiver, except in the case when the receiver already knows as much as the sender. If this event occurs in the source packet transmission phase, mobile terminals having only one source packet missing from the previous packets will be able to recover this packet, which greatly reduces the ordered delivery delay of this packet and all its subsequent ones.

C. PDR Control

The achieved reduction in PDR, from a rate-based proactive retransmission approach, depends on the number of coded packets employed for packet recovery. Obviously, the greater this number, the greater the reduction in PDR. However, increasing the number of retransmission packets will reduce the number of source packets in the block, thus reducing the system throughput.

In [25], we employed a systematic precoding scheme in improving unicast throughput for OFDMA systems with symbol error rate constraints. The same approach can be employed in our multicast application, to compute the minimum value of K that guarantees the reduction of the PDR of any receiver below the target value P_t. For that, it is sufficient to guarantee that the overall PDR of the worst channel mobile terminals (that we will denote by R_w) is below P_t. We will denote the end-to-end packet loss rate of the worst case receiver by P_w. This worst case end-to-end packet loss rate P_w represents the effect of loss probabilities on both uplink and downlink, assuming worst-cast sending and receiving terminals.

For SNC, the overall probability (denoted by P_l) of loosing a source packet at R_w can be expressed as:

$$P_l = P_w \cdot \sum_{j=N-U}^{N-1} \binom{F-1}{j} P_w^j \left(1-P_w\right)^{N-j-1} \qquad (4)$$

In words, a source packet is lost at R_w, if it is lost in the uncoded transmission phase and more than $N-U-1$ packets are lost from the remaining $N-1$ ones. Assuming that packet loss events are independent of each other, we can use the central limit theorem (CLT) to approximate this probability as:

$$P_l \approx P_w \cdot Q\left(\frac{(1-P_w)(F-1)-U}{\sqrt{(F-1)(1-P_w)P_w}}\right) \qquad (5)$$

Since the Q-function is a decreasing function with the increase of its argument, the lower U, the lower P_l. Consequently, the maximum value $U^*(N)$ for a block of N packets, satisfying the PDR constraint, can be computed from (5) as follows:

$$\begin{aligned} P_l \approx P_w \cdot Q\left(\frac{(1-P_w)(1-N)-U}{\sqrt{P_w(1-P_w)(N-1)}}\right) \leq P_t \\ \Rightarrow \quad U^*(N) = \Big\lfloor \left(1-P_w\right)\left(N-1\right) \\ - \theta\sqrt{P_w\left(1-P_w\right)\left(N-1\right)} \Big\rfloor \ , \end{aligned} \qquad (6)$$

where $\theta = Q^{-1}(P_t/P_w)$. Finally, we can express the minimum value for K (denoted by $K^*(N)$), satisfying the PDR constraint, as:

$$\begin{aligned} K^*(N) &= N - U^*(N) \\ &= \left\lceil 1 + P_w(N-1) + \theta\sqrt{P_w\left(1-P_w\right)\left(N-1\right)} \right\rceil \end{aligned} \qquad (7)$$

Since in general $P_t << P_w$, θ will always be positive. It is easy to infer that the expression of $U^*(N)$, without flooring, is superadditive, due to the linearity and increasing monotonicity of its first term, and the subadditivity of its second term for $\theta_i \geq 0$. Since the sum of the floor of two functions is less than the floor of their sum, the supperadditivity applies to $U^*(N)$.

The above argument shows that the greater the value of N, the smaller the value of $K^*(N)$, the higher the system throughput. This justifies our approach of network coding across sessions, rather than coding the packets of each session seperately. It also recommends the increase of N, such that the termination of the block reception occurs at a slight period before the deadline, just enough for packet decoding, filtering and passing to upper layers. In other words, the above formula maximizes the system throughput while jointly satisfying the delay and PDR constraints, through the control of N and P_t.

IV. Performance Evaluation

A. Performance Metrics

In this section, we evaluate the performance of our proposed network coding protocol for both transparent (denoted by NC-T) and regenerative (denoted by NC-R) satellite modes, and compare it with the performance of the SR-ARQ and NC-ARQ protocols, explained in [9]. For SR-ARQ and NC-ARQ, we will assume that the terminals and satellite are allowed to perform reactive retransmissions (in response to negative acknowledgments) only once for the packets of any block.

To compare between the different protocols, we employ the following metrics:

- Average packet delay: defined as the average time spent by the correctly received packets from the start of their

TABLE I: Simulation Parameters

Parameter	Value
Channel bit rate (R_b)	40 Mbps
Packet length (L)	1000 bits
Packet transmission time T_P	25 μs
Timeslot duration	5 ms
Round trip time (RTT)	250 ms

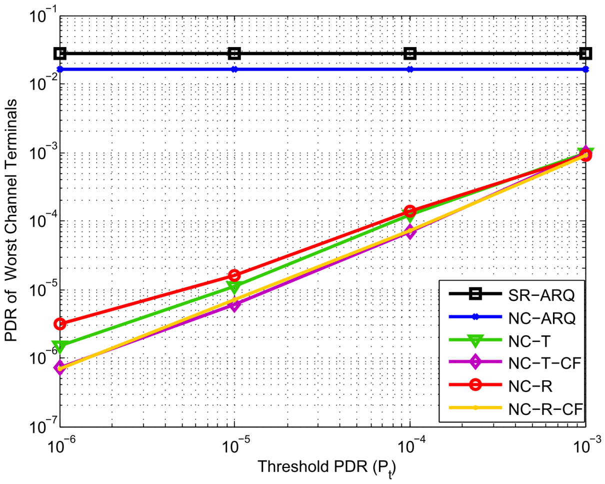

Fig. 3: Worst case PDR of different protocols vs target PDR

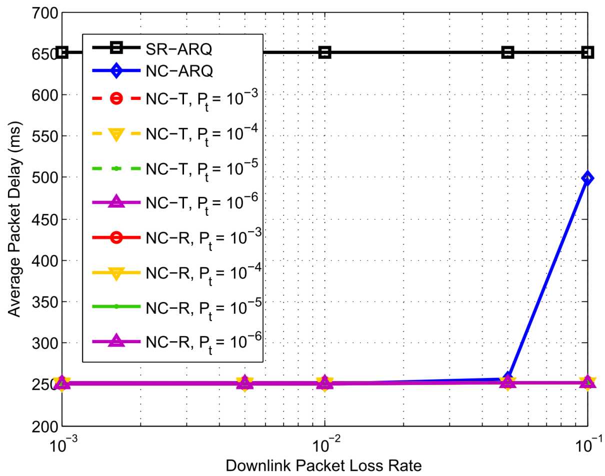

Fig. 4: Average packet delay of different protocols vs packet loss rate

transmission at the sender until their ordered delivery to the upper layers at the receiving terminals.

- Packet drop rate: defined as the ratio of the number of packets that were discarded at the mobile terminals, due to their loss over the system, to the total number of packets that arrived to the sender for transmission.
- Average transmission throughput: defined as the rate at which source packets are transmitted from the sending terminal.
- Average Goodput: defined as the rate at which correct source data is received at the receiving terminals before filtering.

B. Simulation Results

We assume a simulation scenario with one sender and multiple mobile terminals. Table I illustrates the system parameters employed in our simulations. For this block size, we assume a targeted delay of 260 ms, thus allowing 10 ms for packet processing. The different mobile terminals have variable packet loss rates taking values [0.001, 0.005 0.01, 0.05. 0.1] for both uplink and downlink. The simulations are done with MATLAB over a horizon of 1 billion packets.

Figure 3 depicts the comparisons of worst case PDR between SR-ARQ, NC-ARQ and our proposed protocol, for target PDRs of $[10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$, block size N of 200 packets. The figure shows that SR-ARQ and NC-ARQ cannot satisfy the PDR constraints, and their achieved worst case PDRs are orders of magnitude far from the desired target values. We can also see that our proposed protocol is able to tract the target PDR for both transparent and regenerative modes, with a slight degradation resulting from the employed approximation in (5), (6) and (7). To overcome this problem, we introduce empirically calculated correction factors (CFs) that increase the number of network coded packets (in (7)) in each block by 1 or 2 packets. Simulation results show that the proposed protocol, with correction factor, satisfies the PDR constraints for both modes. For the rest of this section, we employ the values of $U^*(N)$ and $K^*(N)$ computed using the empirically calculated CFs that satisfy the PDR targets.

Figure 4 depicts the average delay achieved by the different protocols for the same target PDRs and block size as the previous figure, and for 100 mobile terminals. The performances of all mobile terminals having the same downlink packet loss rate are aggregated in one point for each protocol. The figure clearly shows that SR-ARQ fails to satisfy the delay constraint. For the mobile terminals with better downlink channel quality, the delay deadline is still violated due to potential higher packet loss rate in the uplink. Even for lower packet loss rate in the uplink, the delay does not reduce much, except when the probability tends to zero. For mobile terminals with moderate and worse channel qualities, the deadline is always violated even when $P_i^u = 0$, as has been shown in Figure 3 in [9]. These results can be explained by the very long time between a packet transmission and its retransmission, if lost by any terminal. Consequently, this terminal has to wait for one RTT to obtain another version, which greatly increases the delay. This delay also extends to all packets that follow this first lost packet, as they cannot be passed to upper layers before it.

NC-ARQ provides satisfaction of the delay constraint for mobile terminals with good and moderate channel qualities. This result is achieved due to the selection of the number of proactive retransmission packets according to the packet loss rate of the worst case receiver. This provides a much larger number of such packets than that needed by these good and moderate channel terminals, thus helping them to recover all their packets within the block duration. However, NC-ARQ

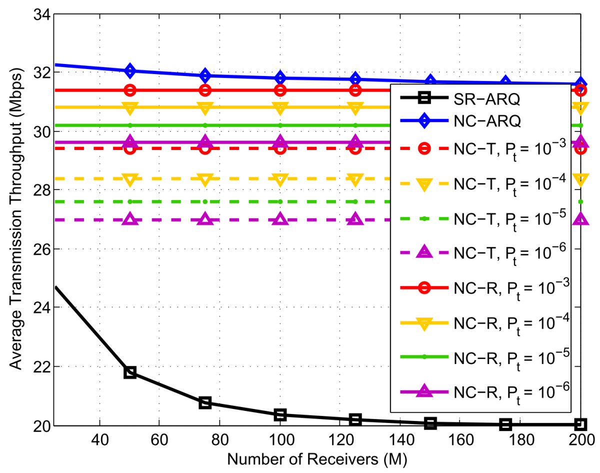

Fig. 5: Average throughput of different protocols vs number of mobile terminals

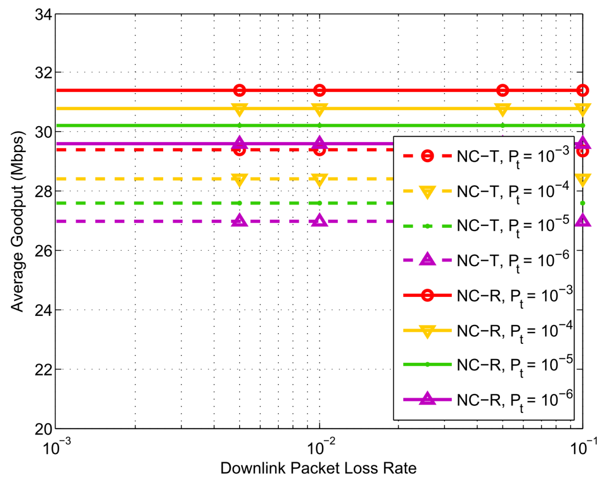

Fig. 6: Average Goodput of our proposed protocol vs the downlink packet loss rate

fails to satisfy the deadline for the mobile terminals with worst channel quality. This can be explained by their increasing dependence on the reactive retransmissions to decode the missing packets in each block.

Finally, our proposed SNC protocol shows very good satisfaction of the delay constraint for all target PDR. This result is trivial due to the structure of the proposed protocol design.

Figure 5 depicts the average transmission throughput performance of different protocols against the number of mobile terminals for the same target PDRs and block size as the previous figures. It can be easily seen that SR-ARQ suffers from larger throughput degradation with the increase in the number of mobile terminals. This can be simply explained by the fact that, the larger the number of mobile terminals, the larger the diversity in lost packets, the larger the number of packets consumed in retransmissions. For more than 100 mobile terminals, the figure shows that every second round is fully wasted in packet retransmissions, which cuts down the system throughput to half the capacity. NC-ARQ demonstrates very good throughput performance for small receiving terminal populations, but slightly degrades as the number of mobile terminals increases. Indeed, the larger the number of mobile terminals, the larger the request for reactive retransmission packets, the lower the throughput.

Our proposed SNC protocol exhibits constant performance regardless of the number of mobile terminals. This is a very good property for network scalability. This performance is obtained since our proposed protocol design fixes the number of coded packets to one value, which is calculated to satisfy the PDR constraint. We can also observe that the degradation in throughput, due to the decrease in the target PDR level, is considerably small, especially in the light of the achieved PDR gains. This throughput reduction can be considered negligible, if compared to a time diversity protocol of diversity order two, which cuts down the capacity to its half, while achieving a much lower PDR reduction.

Another observation is that the NC-ARQ protocol achieves a slightly better throughput than our proposed protocol. However, our protocol have much better control and performance on both worst case delay and PDR, at the expense of this small throughput degradation. This makes our protocol more interesting for a wide spectrum of applications. Finally, we can see that the satellite regenerative mode achieves a considerable improvement in the throughput. This result is achieved due to the replacement of defective packets with correct and potentially innovative packets for all the receiving terminals.

Figure 6 depicts the average goodput performance of our proposed protocol. Similar to Figure 4, the performances of all mobile terminals having the same downlink packet loss rate are aggregated in one point for each protocol. Comparing the results with Figure 5, we can clearly see that the transmission throughput is almost equal to the received goodput, with an unnoticeable degradation representing the PDR.

Finally, Figure 7 depicts the average transmission throughput achieved by the different protocols, when the block size is changed. In this case, the sending terminal encodes different sessions separately and transmit several blocks within on timeslot. The number of mobile terminals in this simulation is 100. We can clearly see that the smaller the block size, the average transmission throughput degrades considerably, which conforms with the superadditivity property of $U^*(N)$, explained in Section III-C. This results justifies the use of inter-session packet mixing using network coding to generate a overall lower number of coded packets. It also supports our recommendation of selecting the block size, so that the termination of the block reception occurs at a slight period before the deadline, just enough for packet decoding, filtering and passing to upper layers.

V. Conclusion and Future Work

In this paper, we proposed a joint delay and packet drop rate control protocol over lossy mobile satellite channels,

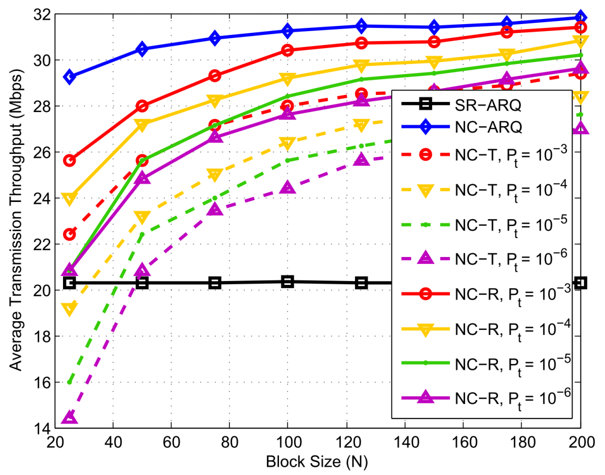

Fig. 7: Average throughput of different protocols vs number of mobile terminals

using network coding. The proposed protocol employs random network coding at the sending terminals, to generate efficient proactive retransmission packets. It also exploits the over-hearing property to combine packets from different sessions in order to generate a smaller number of coded packets, thus increasing the throughput. Moreover, it exploits the fact that random network coding is composable, in allowing the satellite to transmit random network coded combinations of all previously received packets, to fill the gaps left by lost packets on the uplink. By adjusting the timing of the network coded transmissions at the sending terminals, the protocol can control packet recovery at any desired delay, above one system round trip time. The protocol can also control the PDR level, by adjusting the network coding rate using a simple formula and a small correction factor. The protocol is able to achieve these gains at a much higher system throughput, compared to conventional ARQ protocols. Moreover, the protocol enjoys a good throughput scalability for larger network sizes.

In our future work, we will consider the problem of joint delay and PDR control, using network coding, in satellite environments with loss event correlation. We will employ the loss correlation parameters to statistically control the PDR for given delay constraints, such that the probability of PDR constraint violation is minimized. We will also study the potential gains of network coding packets coming from different sending terminals at the satellite. Moreover, we will extend the work to more complicated satellite scenarios, in which network coding has even greater potentials for improvement.

References

[1] H. M. de Lima and O. C. M. B. Duarte, "A point-to-multipoint ARQ scheme with multicopy retransmission for high speed satellite communications," *IEEE International Telecommunication Symposium (ITS'94)*, pp. 357–361, Aug. 1994.

[2] ——, "Point-to-multipoint SR ARQ scheme with accumulative acknowledgment for satellite communications," *Journal of Computer Network and ISDN Sysems*, vol. 30, no. 14, pp. 1311–1325, 1998.

[3] P. Yu and S. Lin, "An efficient selective-repeat arq scheme for satellite channels and its throughput analysis," *IEEE Transactions on Communications*, vol. 29, no. 3, pp. 353–363, Mar 1981.

[4] G. Cao and Y. Wu, "Reliable multicast via satellites," *International Conference on Information Technology: Coding and Computing*, pp. 183–188, Apr 2001.

[5] E. Weigle, M. Hiltunen, R. Schlichting, V. Vaishampayan, and A. Chien, "Peer-to-peer error recovery for hybrid satellite-terrestrial networks," *Sixth IEEE International Conference on Peer-to-Peer Computing (P2P'06)*, pp. 153–160, Sept. 2006.

[6] D. Friedman and A. Ephremides, "A scheme to improve throughput for ARQ-protected satellite communication," *Fifth International Mobile Satellite Conference*, pp. 39–43, Jun. 1997.

[7] S. Cho, "Adaptive error control for hybrid (satellite-terrestrial) networks," *IEEE Wireless Communications and Networking Conference (WCNC'99)*, vol. 2, pp. 1013–1017, 1999.

[8] N. Celandroni, E. Ferro, G. Giambene, and M. Marandola, "Sat01-3: TCP performance in a hybrid satellite network by using ACM and ARQ," *IEEE Global Telecommunications Conference (GLOBECOM'06)*, pp. 1–6, December 2006.

[9] S. Sorour and S. Valaee, "A network coded ARQ protocol for broadcast streaming over hybrid satellite systems," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'09)*, pp. 1–5, September 2009.

[10] U. Mudugamuwa, M. Karaliopoulos, R. Tafazolli, and B. Evans, "Reliable multicast transport and power scheduling for MBMS delivery over 3G mobile satellite systems," *IEEE 59th Vehicular Technology Conference (VTC'04-Spring)*, vol. 5, pp. 2836–2841, May 2004.

[11] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551 –2567, june 2006.

[12] M. Luby, "Lt codes," *IEEE Symposium on Foundations of Computer Science (FOCS'02)*, pp. 271–282, Nov. 2002.

[13] J. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," *IEEE International Symposium on Information Theory (ISIT'08)*, pp. 1651–1655, July 2008.

[14] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," *International Symposium on Information Theory (ISIT'05)*, pp. 1853–1857, Sept. 2005.

[15] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *ACM SIGCOMM'06*, vol. 36, no. 4, pp. 243–254, 2006.

[16] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Médard., "The importance of being opportunistic: Practical network coding for wireless environments," *Allerton*, 2005.

[17] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[18] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communications*, vol. 1, no. 1, pp. 3–20, March 2008.

[19] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," *Fourth Workshop on Network Coding, Theory and Applications (NetCod'08)*, pp. 1–6, Jan. 2008.

[20] A. Yazdi, S. Sorour, S. Valaee, and R. Kim, "Optimum network coding for delay sensitive applications in wimax unicast," *IEEE Conference on Computer Communications (INFOCOM'09)*, pp. 2576–2580, April 2009.

[21] J. Jin and B. Li, "Adaptive random network coding in WiMAX," *IEEE International Conference on Communications (ICC'08)*, pp. 2576–2580, May 2008.

[22] J. Jin, B. Li, and T. Kong, "Is random network coding helpful in WiMAX?" *IEEE Conference on Computer Communications (INFOCOM'08)*, pp. 2162–2170, April 2008.

[23] F. Vieira and J. Barros, "Network coding multicast in satellite networks," *5th Euro-NGI conference on Next Generation Internet Networks (NGI '09)*, pp. 1 –6, July 2009.

[24] F. Vieira, M. Castro, and J. Lei, "Datacast transmission architecture for dvb-s2 systems in railway scenarios," *10th International Workshop on Signal Processing for Space Communications (SPSC'08)*, pp. 1 –5, Oct. 2008.

[25] S. Sorour, A. Yazdi, S. Valaee, and R. Kim, "Throughput improvement through precoding in ofdma systems with limited feedback," *IEEE International Conference on Communications (ICC'09)*, pp. 1 –5, June 2009.