Raptor-Network Coding Strategies for Energy Efficient Cooperative DVB-H Multimedia Communications

Lucien Benacem
Department of Electrical & Computer Engineering
Queen's University
Kingston, Ontario, Canada, K7L 3N6
lucienbenacem@ieee.org

Steven D. Blostein
Department of Electrical & Computer Engineering
Queen's University
Kingston, Ontario, Canada, K7L 3N6
steven.blostein@queensu.ca

***Abstract*—Reliable and energy-efficient delivery of mobile multimedia across different platforms and application scenarios is very challenging. Both open and proprietary standards exist worldwide; we focus on enhancements to the digital video broadcasting – handheld (DVB-H) standard for transmission over *cooperative* cellular networks, which enables the use of relaying to supplement existing fixed infrastructure. Cooperation creates a distributed form of multi-input-multi-output (MIMO) systems, resulting in breakthroughs in network energy efficiency and reliability. Thus, cooperative networks have an inherently green ad hoc topology that is able to substantially reduce infrastructure cost, system complexity and energy expenditure. In this paper, novel strategies are proposed and evaluated that combine: (1) fountain coding, e.g., Raptor codes at the application layer, and (2) network coding. Originally proposed for broadcasting over the Internet, the application of fountain codes to wireless cooperative communications networks has been limited to date. These codes enable lower (physical) layer compatibility. Network coding is used to reduce energy consumption by opportunistically recombining and rebroadcasting required combinations of packets. The energy cost of our peer-to-peer file repair sessions, which are integrated with the file delivery session itself, are quantified. Signal processing and modeling techniques used for cross-layer system simulations are also overviewed.**

I. INTRODUCTION

In broadcasting multimedia to many users, many energy- and reliability-related challenges arise at the lowest layers of the protocol stack, such as the physical and data link layers. However, to avoid large upfront infrastructure costs, network operators generally tend to reuse as much of an existing network infrastructure as possible. For example, the DVB-H standard relies heavily on the terrestrial DVB-T infrastructure. Therefore, the optimization potentials on the layers below the network layer are very limited from a practical perspective and Internet Protocol (IP) broadcast transmission is generally not fully optimized, necessitating higher-layer protection schemes. Application-layer forward error correction (AL-FEC) fountain codes such as Raptor codes [1] have thus grown in popularity due to desirable properties that include capacity-approaching performance on erasure channels, scalability, coding efficiency and the fact that they can be implemented in software. For wireless applications, Raptor codes are preferred over Automatic Repeat reQuest (ARQ) for error control. Network coding (NC) has also evolved as a method of increasing the throughput of a network and can be combined with Raptor codes.

This research is supported by the Natural Sciences and Engineering Research Council on Canada Grant STPSC 356826-07.

DVB-H provides an efficient standard for carrying multimedia over digital terrestrial broadcasting networks to handheld terminals. It introduces two main innovations over DVB-T; time sliced burst transmission (Fig. 1) and data link layer FEC known as multi-protocol encapsulation FEC (MPE-FEC). DVB-IPDC (IP Datacast) combines a unidirectional point-to-multipoint broadcast path with a bidirectional interactivity path. In this work we focus on the former. A simplified protocol stack for a DVB-IPDC system is presented in Fig. 2. The physical and data link layers define a DVB-H system, and the network, transport and application layers define a DVB-IPDC system. Application layer Raptor codes have been mandated for use in DVB-H for file download applications using the File deLivery over Unidirectional Transport (FLUTE) protocol. In Fig. 1, P_{RF_ON} is the power consumed while the RF and demodulation parts of the receiver are active, $P_{RF\_OFF\_1}$ is used while the RF part is off but MPE-FEC is ongoing, $P_{RF\_OFF\_2}$ is used while data is fed to the application layer, and $P_{RF\_OFF\_3}$ is used while the receiver waits for the next burst.

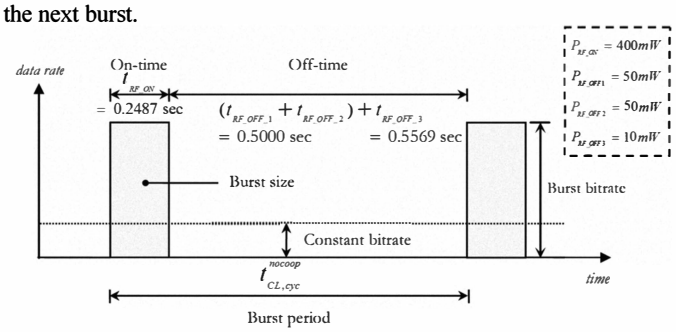

Figure 1: Time-sliced burst characteristics

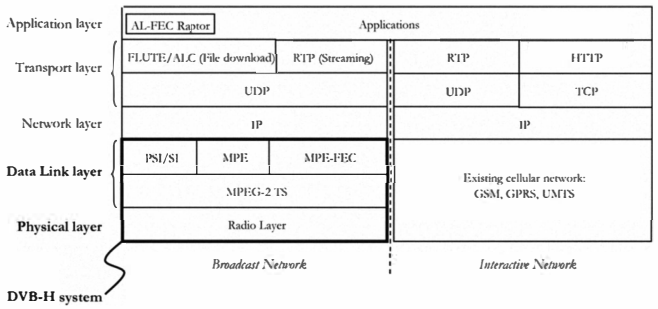

Figure 2: Simplified DVB-IPDC protocol stack

The basic idea of cooperative DVB-H reception was initially presented in [2]. The idea is that terminals can cooperatively receive time-sliced bursts from the base station (BS). Each cooperative terminal only receives a fraction of the total data transmitted by the

BS over the cellular link (CL) each $t_{CL,cyc}^{coop}$ time period. In the case where multiple services are transmitted within a burst, the terminal does not discard the unwanted packets, as would be done in the current implementation of DVB-H. Instead, the terminal forwards those packets to its neighbouring terminals over its short-range link (SRL) — it is assumed that each terminal has multiple radio interfaces, which is realistic of most mobile phones today. Based on reciprocity, the device receives the missing packets it needs from its neighboring terminals. We refer to this strategy as Cooperative Strategy 0 (CS0) going forward.

In this paper, we examine the intersection of cooperative communications, Raptor AL-FEC, and opportunistic NC. We first extend the current DVB-H standard to support cooperation among terminals, forming a cooperative broadcast network (CBN). Then, we develop two novel low-complexity cooperative schemes for time-insensitive file delivery sessions that can be used in today's DVB-H networks to simultaneously and significantly improve both the reliability and energy efficiency of multimedia communications.

II. System Model

A. DVB-H/DVB-IPDC Dynamic System Simulator

For our investigations, we developed a discrete-event DVB-H dynamic system simulator in MATLAB & Simulink, shown in Fig. 3.

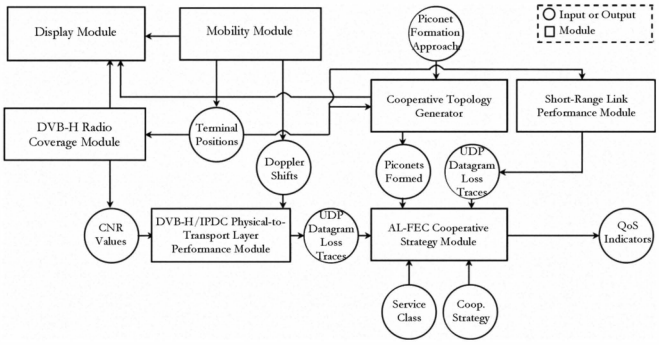

Figure 3: DVB-H/DVB-IPDC dynamic system simulator

The main components of the simulator are:

- **Terminal mobility module**: terminals move according to a correlated Gauss-Markov process with memory factor of 0.5. The minimum terminal speed is 1m/sec and the maximum is 2m/sec. The maximum pause (stationary) time of a terminal is 20 sec.
- **DVB-H radio coverage module**: combines large scale Okumura-Hata path loss, log-normal shadowing and Rayleigh fading
- **DVB-H/DVB-IPDC physical-to-transport layer performance module**: based on a hierarchical Markov model that performs a complete mapping from physical layer transport stream (TS) packet errors to the symbol losses seen at the application layer.
- **Short-range link performance module:** based on the Bluetooth (BT) 2.1 + EDR standard power class 1 as described in [3].
- **Cooperative topology generator:** forms Bluetooth piconets
- **AL-FEC Cooperative Strategy module:** applies strategy rules
- **Display module:** provides a visual display of network dynamics

The interested reader is referred to [4] for an in-depth description of the simulator and the parameters used for each component. We note that the SRL BT radio uses 100mW of RF power over 0.1245 sec to transmit a time-sliced burst, 10mW of power to receive a burst over 0.1245 sec, and 1mW of power in its idle state in the remaining time.

B. Simulated Scenario

We consider a single outdoor cell of a DVB-H single frequency network, with a BS located at the centre. 100 terminals are initially uniformly distributed across the circular cell, which has radius 2.1 km — a user density corresponding to the initial adoption of DVB-H in a service area. Each terminal is transported by a slow-moving or stationary pedestrian user ("Class A" terminal). Terminals opportunistically form ad-hoc piconets with their neighbours when they are physically close enough, following a selfish energy-minimization strategy which favours groups of 2 or 3. We consider only fully-meshed piconets. To better understand the network dynamics, the joint pdf of piconet lifetime and piconet size are shown in Fig. 4, from which it can be seen that the average piconet lifetime is 33.19 seconds and the average size is 2.08 terminals. The base station continuously transmits time-sliced bursts for the full duration of a file delivery session, where a 500KB file is sent across 36 bursts.

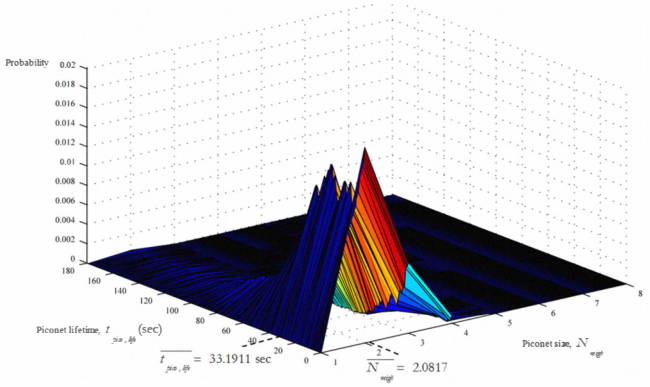

Figure 4: Joint pdf of network dynamics

Using the dynamic system simulator, we were able to observe User Datagram Protocol (UDP) losses resulting from the physical conditions of the network as it varied through time. Thus, from the perspective of AL-FEC and NC, the simulator produced the output of a Binary Erasure Channel with time-varying erasure probability or loss rate $p(t)$, dependent upon the lower protocol layers, as in Fig. 5.

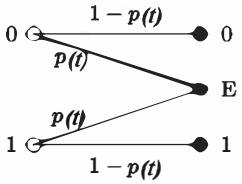

Figure 5: Binary Erasure Channel with time-varying loss rate

III. Non-Systematic Raptor Codes

Raptor codes are an extension to Luby Transform (LT) codes. A Raptor code with parameters $(K, C, \Omega(x))$ is an LT code with an (n, K) linear code C as a pre-code (outer code) and a distribution $\Omega(x)$ known as the Robust Soliton Distribution (RSD) on n input symbols that are the coordinates of codewords of C. The precode is described by an $n \times K$ generator matrix G. The key idea of Raptor coding is to relax the requirement that all of the source symbols must be recovered. In doing so, an LT code is must only recover a constant fraction of the source symbols. The decoding graph for a non-systematic Raptor code is shown in Fig. 6. n input symbols are obtained by pre-coding the K source symbols with the (n, K) block code. Next, the LT inner code is used to produce a potentially infinite stream of output symbols z_j. Once the decoder has $N > K$ output symbols, the source symbols can be recovered with high probability.

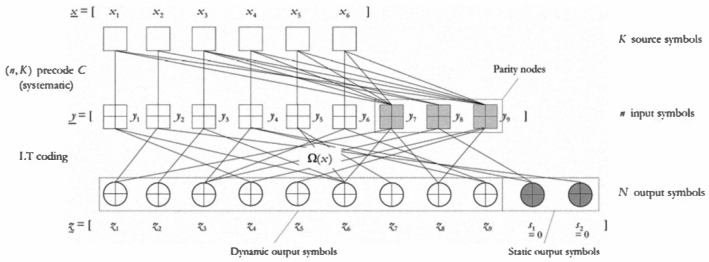

Figure 6: General decoding graph of non-systematic Raptor code

IV. SYSTEMATIC STANDARDIZED RAPTOR CODES

Fig. 7 shows a conceptual schematic of the entire encoding process for a systematic Raptor code. The innermost code is an LT code, which provides the rateless property of the overall code. The LT encoder in combination with a systematic precode forms the non-systematic Raptor encoder. Finally, a transformation on the source symbols (via matrix R) provides the systematic property of the overall code.

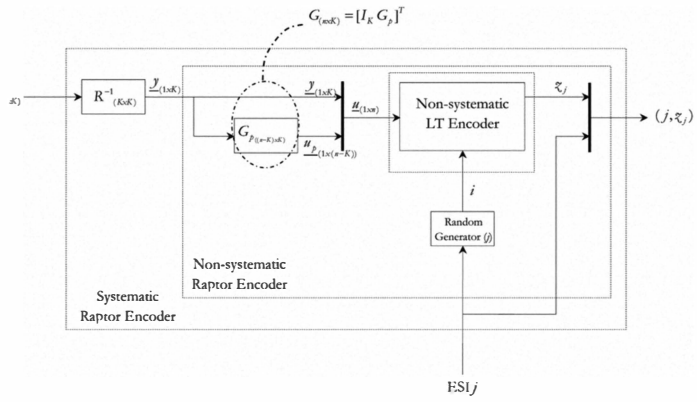

Figure 7: Conceptual schematic of systematic Raptor encoder.

Assume that the Raptor code has a reliable decoding algorithm of overhead ς symbols. Second, the precode C is systematic and has an $n \times K$ generator matrix $G = [I_K \; G_p]^T$, where I_K is a KxK identity matrix and G_p is an $(n-K)\times K$ submatrix corresponding to the parity information in a precode codeword. Each output symbol is obtained by sampling independently from $\Omega(x)$ in the LT encoder to obtain a row vector $\underline{v}$ in $\mathbb{F}_2^*$ and the value of the output symbol is calculated as the scalar product of pre-coded intermediate symbols (PISs) $\underline{u}$ with $\underline{v}$, i.e. $z_j = \underline{v} \cdot \underline{u}^T$. For a given set N of output symbols, there is an $N \times n$ matrix S for which the rows are the vectors corresponding to the output symbols. Then, we have that the overall encoding process is:

$$S \cdot G \cdot \underline{x}^T = \underline{z}^T. \qquad (1)$$

Decoding the Raptor code is equivalent to solving the system of equations in Eq. (1) for $\underline{x}$, and is done in two steps:

1. Decode the output symbols using LT-decoding to obtain the PISs $\{u_1,...,u_n\}$, then extract $\{y_1,...,y_K\}$. If $\{u_1,...,u_n\}$ cannot be recovered, a decoding error has occurred.
2. Calculate $\underline{x}^T = R \cdot \underline{y}^T$.

Raptor codes in DVB-H are decoded using a maximum likelihood Gaussian elimination algorithm. The decoding algorithm was designed such that once the PISs $\underline{u}$ are found, the source symbols $\underline{x}$ can be computed very efficiently. The standard's recommended algorithm is based on the concept of a "code-constraint processor" and a code-constraint matrix $A_{(N\times n)}$, which provides a set of constraints for both the pre-coding *and* inner LT coding and is described fully in [5]. After receiving r output symbols with unique encoded symbol IDs (ESIs), the code-constraint matrix can be written as $A(j_1, j_2, \ldots, j_r) \cdot \underline{u}^T = [z_1, z_2, \ldots, z_r]^T$. Once $rank(A) = n$, $\underline{u}^T$ can be fully recovered. Since the inner LT code's generator matrix, G_{LT}, is a sub-matrix of A, its rank is critical to the overall rank of A. In Sec. VIII we examine $rank(G_{LT})$. The decoding algorithm, which is comprised of 4 stages, can be implemented in real-time even with limited computing capability for message lengths in the range $4 \le K \le 8{,}192$.

The FLUTE protocol splits a file into multiple source blocks. Each source block consists of K source symbols (each s_{sym} bytes) and is encoded using the standardized Raptor code independently of the other blocks. Each resulting encoded symbol is assigned a unique ESI and one or more symbols are then grouped consecutively into encoded packets. FLUTE packets are encapsulated into UDP datagrams and are distributed over the IPDC broadcast bearer. The BS transmits a fixed amount of redundancy (repair symbols) after the source symbols are transmitted based on its estimate of the worst CL burst loss rate experienced by a terminal. For the 500KB file under consideration, $K = 1000$ symbols. We assume a 10% fixed overhead, giving a total of $N = 1100$ symbols. Note that this fixed redundancy is not the coding reception overhead ς. In DVB-IPDC systems, if a file recovery fails, a post-delivery repair phase is invoked. In this work, we do not consider the energy costs of a post-delivery repair session.

V. COOPERATIVE STRATEGY 1: MULTICAST

First, we address the special case of *multicast*, where each neighbour in a cooperative piconet desires the same service from the BS. In our investigations, we adopted a link layer decoding scheme known as section erasure decoding, whereby an entire MPE-FEC frame (burst) is either decoded completely correctly or incorrectly, i.e. it is lost or erased. Fig. 8 shows a timing diagram of strategy CS0-M as proposed in [2]. The top sub-graph shows the BS transmission pattern in the TS rate-time domain, while the other subgraphs show the transmission and reception of bursts by each terminal according to the cooperative round-robin transmission schedule. Each BS burst is comprised of multiple elementary streams (services), but all terminals are interested in service S_2. The terminal receiving the CL burst during each non-cooperative cyclic time period $t_{CL,cyc}^{coop}$ is known as the *intermediate* terminal, and those receiving their data over the SRLs are *destination* terminals.

We observe that in the second time period, terminal *T2* is unable to properly decode the burst from the BS. As a result, all other N_{neigh} neighbours in the cooperative piconet experience a *propagated burst loss* over the SRL. A *channel burst loss*, on the other hand, results from a poor channel that leads to a burst erasure. To see the severity of this problem if CS0-M were applied to our setup, we quantified the average cumulative percentages of each of the two types of SRL losses over 100 independent file delivery sessions using the dynamic system simulator of Sec. 2. We determined that (a) cooperative terminals using the transmission schedule of Fig. 8 always experience more total loss than non-cooperative terminals, as expected, and (b) that the propagated SRL losses comprise a non-negligible fraction of the total, typically accounting for about one third [4] of all losses experienced by a cooperative terminal. For this reason, without further modifications, CS0-M would be unsuitable for use in a practical (lossy) DVB-H system.

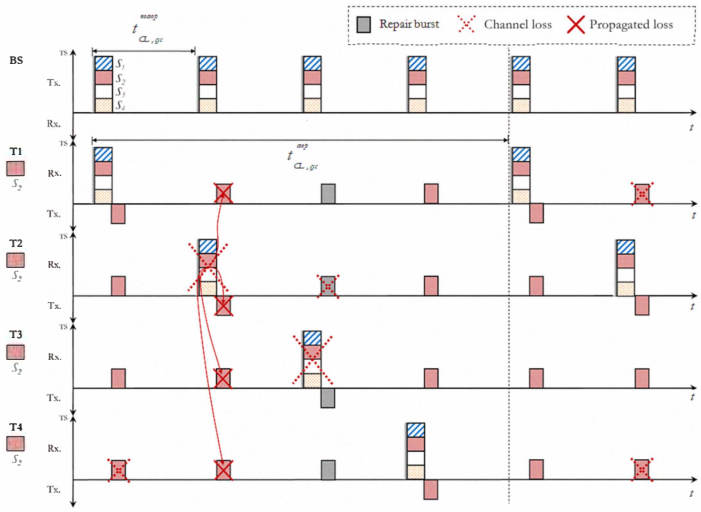

Figure 8: CS0-M/CS1-M timing diagram ($N_{neigh} = 4$)

To improve the performance of CS0-M while keeping its desirable energy saving quality, we developed scheme CS1-M. Referring again to Fig. 8, in time period 3 terminal $T3$ drops its corrupted CL data in the third burst. Next, it generates a set of repair symbols $\underline{r}$ from its current memory buffer of decoded PISs $\underline{u}$, by LT-encoding some of them together. Note that we implicitly assume that each terminal uses a greedy Raptor decoder. $T3$ then transmits a burst consisting of the generated repair symbols over its SRL. In this manner, the causative loss relationship of CS0 is removed and the uniformity property of the systematic Raptor code is exploited in order to provide new information that is useful to all neighbours in the piconet and advances the decoding of their source blocks.

Between time $t_0 = 0$ sec and the time of the first CL channel burst loss in the cell, the BS broadcasts a total of $q \le N$ symbols. The partial set of output symbols transmitted by the BS is then $\underline{z}_{partial} = \{z_1, z_2, \ldots, z_q\}$ and all terminals Ti within the cell collect subsets of $\underline{z}_{partial}$, $\underline{z}_{partial,Ti}$. The sets $\underline{z}_{partial,Ti}$ differ both in size and contents due to the heterogeneous reception conditions over the CL and SRLs. Let $\underline{z}_{partial}$ be decodable to a partial set of PISs $\underline{u}_{partial} = \{u_1, u_2, \ldots, u_a\}$, where $a \le n$ .Then every cooperative terminal possesses a subset of $\underline{u}_{partial}$, corresponding to the decoded z_i in its $\underline{z}_{partial,Ti}$. Upon experiencing a CL burst loss, an intermediate terminal draws upon its set $\underline{u}_{partial,Ti}$ to generate a *repair burst*, and subsequently forwards the burst over the SRL to its neighbours. For this on-the-fly method recoding scheme to work, we first provide solutions to two related problems, which we call the *symbol availability problem* and the *symbol sampling distortion* problem.

A. The Symbol Availability Problem

A typical Raptor code decoding characteristic follows the well-known "avalanche" decoding behaviour as shown in Fig. 9, whereby very few input symbols are decoded until a critical number of received output symbols is reached, at which point nearly all of the input symbols are recovered. This implies that very few decoded PISs would be available to a terminal for recoding until the entire file had already been transmitted. To solve this problem, we introduce a simple *pre-processing* step at the BS which takes place prior to transmission. Our algorithm results from two important observations; (1) Raptor encoding in DVB-H is *not* performed in a rateless way, nor is it done on-the-fly by the BS as assumed in theoretical Raptor code discussions. Instead, a fixed overhead is computed *prior* to the start of the file delivery session according to the BS's estimate of the worst CL channel as in Sec.4.This is logical from a practical perspective because it prevents a few terminals experiencing high loss rates from causing an excessively long file delivery session whereby the BS's resources are not being used efficiently. (2) We note that the cause of the steep decoding characteristic of Raptor codes is primarily due to the fact that output symbols are transmitted in a *random* order. This results in many output symbol encoding dependencies that accumulate and cannot be satisfied until most of the encoded symbols have been received. As a result, the inner LT code of the systematic Raptor code exhibits very poor intermediate decoding performance.

To overcome these problems, we apply a pre-processing step after all N Raptor-encoded symbols corresponding to a file have been generated, but prior to transmission of those symbols by the BS over the channel. After the encoding process is complete, generator matrix G_{LT} exists. The algorithm performs an intelligent re-ordering of the transmission sequence of output symbols such that the degrees of encoded output symbols can be reduced in an efficient, incremental way. The result is greatly improved interim decoding performance as shown in Fig. 9. The full BS pre-processing and terminal re-coding algorithms are presented in [4].

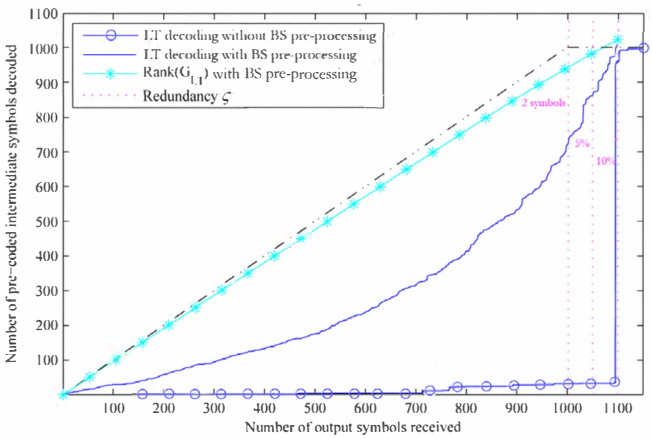

Figure 9: Inner LT decoding characteristic with(out) pre-processing

B. The Sampling Distortion Problem

The second problem facing strategy CS1-M is that the Raptor encoding process normally occurs at the BS, where the *entire* file is available for encoding, rather than at a terminal where only *partial* file information is available. In order to generate new repair symbols, a degree i must be sampled from the RSD, followed by a uniformly random selection of i pre-coded intermediate symbols from the set of *all* n pre-coded intermediate symbols. The availability of only a fraction of the PISs at a cooperative terminal forces the terminal to over-sample PISs decoded earlier in the file delivery session relative to ones decoded later in the session, when more symbols have been decoded and are available. This problem can be thought of as inducing a distortion on the posterior PIS sampling pdf. We assume that each symbol in the file is equally important, since the user satisfaction metric is essentially binary; the file was decoded or it was not. Therefore, in our work this distortion is an undesirable effect and we wish to mitigate it. We note, however, that sampling distortion may be desirable in Unequal Error Protection scenarios where different source symbols have different importances to the quality metric under study. In [4], we present an algorithm that allows the system designer to trade-off an intermediate terminal's ability to produce repair symbols with the resulting PIS sampling distortion introduced by over-sampling certain PISs.

VI. Linear Network Coding over Lossless Channels

Network coding is a technique which allows a terminal to combine a number of incoming packets into one or more outgoing packets, rather than simply forwarding or decoding and forwarding the same packets to the destination terminals. In this way, NC can increase throughput or, alternatively, reduce energy consumption by reducing the number of sent packets to carry the same information.

Consider an original packet $\underline{m}$ of size M bits, consisting of $M/V \in \mathbb{Z}_+$ symbols $\underline{v}$ of size V bits each over $\mathbb{F}_2^V$. With linear NC, outgoing NC packets are linear combinations of Y original packets $\underline{m}_1, \underline{m}_2, \ldots, \underline{m}_Y$, with addition over $\mathbb{F}_2^V$. The original packets may come from multiple terminals. A global encoding vector $\underline{g}_j = [g_1, g_2, \ldots, g_Y]$, dictates which original packets were added together to produce the NC packet $\underline{w}_j = \sum_{i=1}^{Y} g_{j,i}\,\underline{m}_i$. We assume that the encoding vector is sent along with $\underline{w}_j$ as side information in its packet header, and that the overhead in doing so is negligible. $\underline{g}_j$ is used by recipient terminals to decode the data in a NC packet.

VII. Cooperative Strategy 1: Multiple Unicasts

We now consider the special case of *multiple unicasts*, where each terminal in a piconet requests a different service from its neighbours, as shown in Fig. 10. Cooperative energy consumption is necessarily higher in this case for CS0-MU since more SRL transmissions are required. We now present an algorithm that makes use of opportunistic NC to reduce this number. CS1-MU is different from CS1-M in four ways. First, we allow terminals to overhear SRL transmissions intended for their neighbours, a process we call *snooping*. A terminal Ti stores received symbols for its requested service $S_{Q(Ti)}$ in a primary buffer $B_{Ti,Q(Ti)}$, where Q is a mapping function between a terminal and its requested service. Those symbols related to its neighbours' services are kept in secondary buffers $B_{Ti,Q(Tk)}$ $1 \le k, i \le N_{neigh}$, $k \ne i$ (Fig. 11). Second, we introduce the idea of a *buffer update burst*, denoted $U_{Ti,b}$, $b = 1, 2, 3, \ldots$ sent every $t_{buffer\_burst} = \lambda t^{snoop}_{CL,QC}$ period, $\lambda \in \mathbb{Z}_+$. Each terminal maintains for each of its neighbours an *ESI list* describing symbols the neighbour is known to possess in its buffers. Third, after an intermediate terminal Tj experiences a CL loss, a *single* "poisoned" NC repair burst is generated by Tj and transmitted over the SRL to all of its neighbours. Fourth, terminals do not do any Raptor recoding, only network coding of already Raptor-encoded symbols. The full algorithm is presented in Fig.12. A buffer memory management algorithm is also presented in [4].

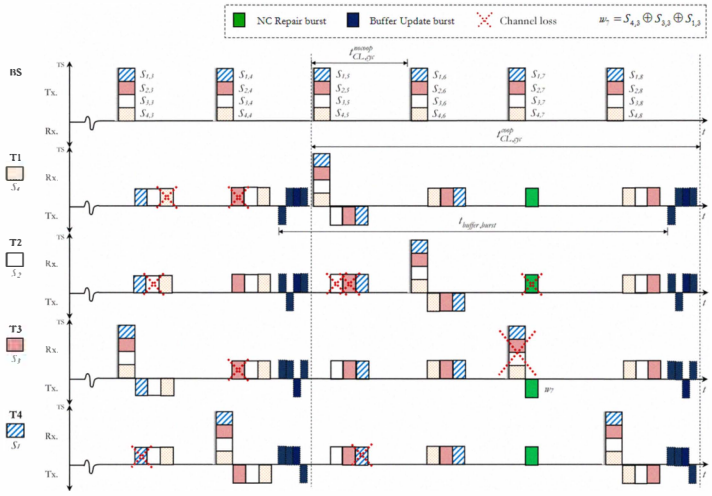

Figure 10: CS1-MU timing diagram ($N_{neigh} = 4, \lambda = 4$)

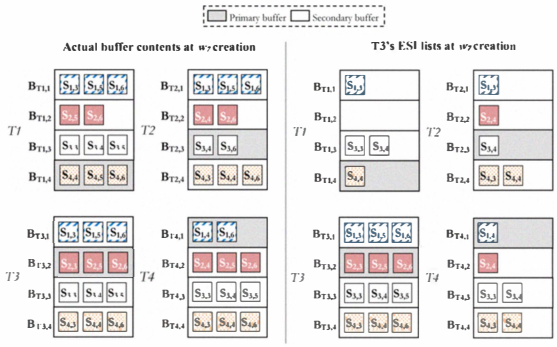

Figure 11: Terminal buffers at time of NC repair burst creation

Note that in Fig. 11, each block represents a number of encoded symbols E with consecutive ESIs corresponding to a certain service.

Upon loss of a CL burst by intermediate terminal Tj :

1. Using its ESI lists, Tj determines which already-seen symbols from the BS are missing from each of its neighbours' primary buffers. These $N_{neigh} - 1$ sets of missing symbols are denoted by $\mathrm{M} = \{M_{T1}, M_{T2}, \ldots, M_{Tk}\}$, where $1 \le k \le N_{neigh}, k \ne j$.
2. ***For each*** $k \ne j$ Tj ***calculates:***

$$M_{Tk}' = M_{Tk} \cap B_{Tj,Q(Tk)} \tag{2}$$

The resulting sets $\mathrm{M}' = \{M_{T1}', M_{T2}', \ldots, M_{Tk}'\}$, $1 \le k \le N_{neigh}$, $k \ne j$ represent the symbols that are missing from Tj's neighbours' primary buffers that Tj is able to provide.

3. ***For each*** $k \ne j$ Tj ***calculates:***

$$M_{Tk}'' = M_{Tk}' \cap \left(\bigcap_{\substack{i=1 \\ i \ne j,k}}^{N_{neigh}} B_{Ti,Q(Tk)} \right) \tag{3}$$

Eq. (3) produces new sets $\mathrm{M}'' = \{M_{T1}'', M_{T2}'', \ldots, M_{Tk}''\}$, $1 \le k \le N_{neigh}, k \ne j$ corresponding to symbols in M_{Tk}' that every neighbour to Tk other than Tj also possesses.

4. Tj selects one symbol $E_{Tk,g}$ $1 \le g \le |M_{Tk}''|$ from each M_{Tk}'' with $|M_{Tk}''| > 0$. Tj then produces a NC repair symbol $E_{rep,g}$ by taking the XOR of the selected symbols as:

$$E_{rep,g} = \sum_{\substack{k=1 \\ k \ne j}}^{N_{neigh}} \oplus E_{T_k,g} \tag{4}$$

and appends $E_{rep,g}$ to the symbols comprising repair burst w_b.

5. ***While*** ($|w_b| \le$ ***maximum number of symbols per burst)***
   - Tj repeats step 4 with subsequent symbols from each M_{Tk}'', appending the new repair symbols to burst w_b
   - The process ends prematurely if $|M_{Tk}''| > 0$ for only one k.
6. Tj transmits w_b over its SRL to its neighbours.
7. ***For each of*** Tj ***'s neighbours*** Tk ***:***

   If (w_b ***is correctly received by*** Tk ***over the SRL)***
   - Each symbol within w_b is decoded using the appropriate "antidote" symbols in Tk's secondary buffers
   - Each decoded symbol is stored in Tk's primary buffer

Figure 12: CS1-MU repair burst generation algorithm

VIII. Results and Analysis

Using the simulator of Sec. II, we evaluated both schemes CS1-M and CS1-MU by comparing them to CS0-M and CS0-MU, respectively for the same BS transmission rate. Energy calculations were done by summing the energy expenditures for the various terminal transmissions and receptions over the CL and SRLs using the parameters in Fig. 1 and Sec. II. All results were averaged over 100 independent pre-processed BS LT generator matrices G_{LT} and then over10 mobility module traces for each matrix. We evaluated the following metrics for *assisted destination terminals*: (1) number of decoded PISs and (2) *rank*(G_{LT}), as functions of file delivery session progression. We also computed cumulative energy consumption as a function of file delivery session progression for *all* cooperative and non-cooperative terminals in the network.

In Fig. 13, we prevented recoding from starting until burst 8. Thus, no symbols were decoded by intermediate terminals up to this point. The average number of PISs decoded under strategy CS1-M at the end of the file delivery session was 683, substantially exceeding the average of 401 decodable under strategy CS0-M. Similarly, the average assisted destination terminal's G_{LT} rank at the end of the session was 947 under CS1-M as compared to 739 under CS0-M. We notice that both sets of curves diverge as the file delivery session progresses because, as the size of an intermediate terminal's set $\mu_{partial,Ti}$ grows, the repair symbols it produces are comprised of more evenly distributed selections of PISs across the whole file. This implies that the resultant columns of G_{LT} at the destination terminals corresponding to the received repair symbols have progressively less linear dependence on existing G_{LT} columns and contribute to greater increases in rank and number of decoded symbols.

Examining Fig. 14, we see that the average energy consumption of CS0-M and CS1-M are very similar, with CS1-M slightly higher than the former due to the SRL repair bursts sent. This small increase in energy, however, is justified based on CS1-M's improved decoding.

In Fig. 15, for CS0-MU, the decoding and rank performances of the average destination terminal do not change appreciably from the case of multicast. The decoding curve finishes significantly higher for CS1-MU than both CS0-MU and CS1-M because the repair symbols provided in this case are a "second chance" at those previously transmitted by the BS, rather than recoded symbols that may have linear dependencies on *Tk*'s existing G_{LT} columns. The energy consumption of CS1-MU is noticeably lower than CS0-MU because of the saved SRL bursts enabled by network coding of Raptor-encoded symbols. Under both schemes, non-cooperative terminals had significantly higher energy consumption than their cooperative counterparts: 55% higher than CS1-M in the multicast case and 100% higher for the multiple unicasts case.

IX. Conclusions

In this paper, we developed two novel coding-based strategies for simultaneously achieving increased energy savings and reliability in a file delivery session. We studied our schemes in the context of a future extension to the DVB-H standard to incorporate terminal-to-terminal cooperation. Our schemes are fully backwards compatible and could be implemented today without making any large changes to existing DVB-H infrastructure and networks. Our results show substantial promise for the future of coding-enabled cooperative multimedia communications.

References

[1] M.A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, 2006, pp. 2551-2567.

[2] Q. Zhang, F. Fitzek, and M. Katz, "On the Energy Saving Potential in DVB-H Networks Exploiting Cooperation among Mobile Devices," Cognitive Wireless Networks, 2007, pp. 473-484.

[3] *Specification of the Bluetooth System: Covered Core Package version 2.1 + EDR*, 2007.

[4] L. Benacem, "Cooperative DVB-H: Raptor-Network Coding Enabled Energy-Saving Strategies" M.Sc (Eng.) thesis, Queen's University, Kingston, ON, Canada, 2010.

[5] M. Luby, M. Watson, T. Gasiba, T. Stockhammer, and W. Xu, "Raptor Codes for Reliable Download Delivery in Wireless Broadcast Systems," *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, 2006, pp. 192-197.

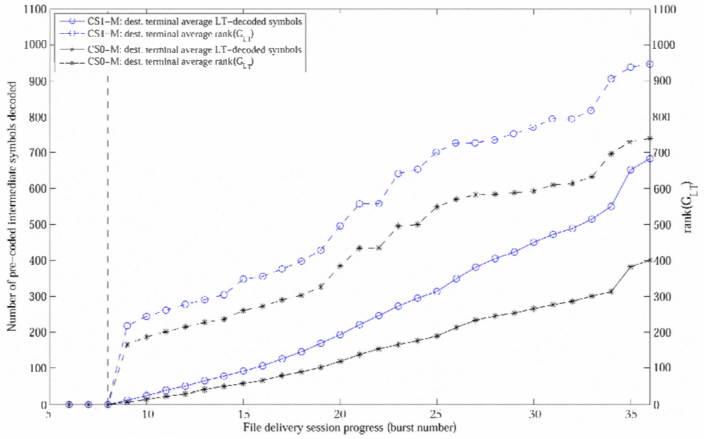

Figure 13: CS1-M vs. CS0-M: LT-decoding/ *rank*(G_{LT}) comparison.

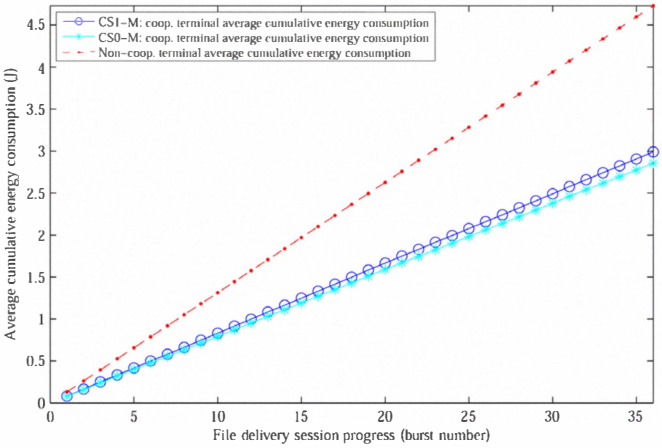

Figure 14: CS1-M vs. CS0-M: cumulative energy use comparison.

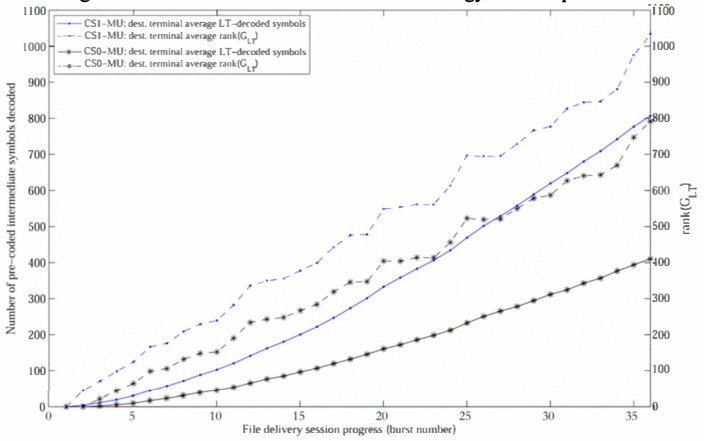

Figure 15: CS1-MU vs. CS0-MU: LT-decoding/ *rank*(G_{LT}) comparison.

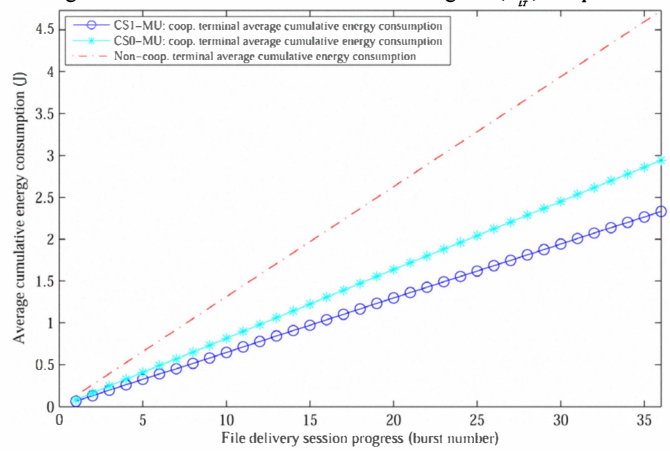

Figure 16: CS1-MU vs. CS0-MU: cumulative energy use comparison.