



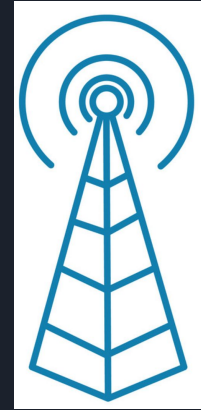
A Randomized Algorithm for Coded Cooperative Data Exchange ^[1]

Written By: Alex Sprintson, Parastoo Sadeghi, Graham Booker, and Salim El Rouayheb

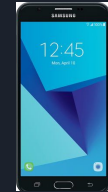
Presented By: Tim Castiglia, Michael Giancola, and Ingkarat Rak-Amnouykit

Problem Description

- Information exchange between wireless clients
- Each client
 - holds a subset of packets
 - knows which packets are available to other clients
 - can broadcast packets it processes
- Broadcasting
 - To every client via noiseless channel.
 - One linear combination of packets each time
- Goal
 - Every client obtains every packet
 - Minimize the total number of transmissions



[3]



[2]

Naïve Solutions

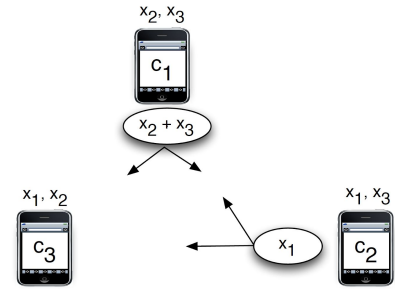


Fig. 1. Coded data exchange among three clients.

- Each client broadcast its packets to others
- One naive solution is to have each client sends uncoded transmissions. In figure 1: $C_1 \rightarrow x_2$ $C_2 \rightarrow x_3$ $C_3 \rightarrow x_1$
- Not always optimal
- With linear coding, C_2 broadcasts x_1 , and C_1 broadcasts a linear combination of $(x_2 + x_3)$

[1]

Linear Coding

- Packets = codewords in a finite field

- \mathbb{F}_q has q elements, q is prime
- Ex: $q=5$, packet=[0 1 4 3 2]

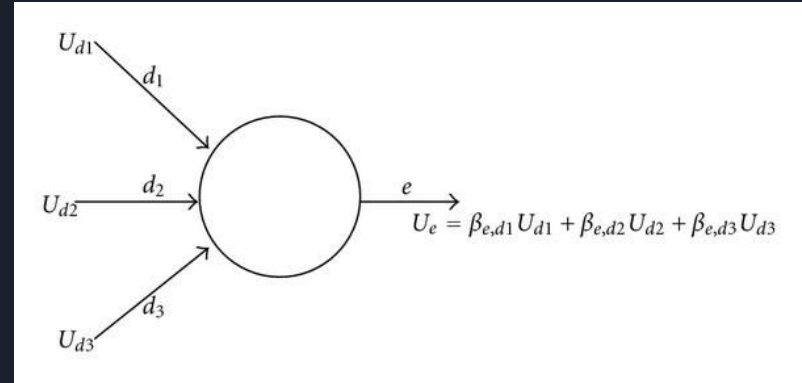
- Linear combination of packets = codeword

- $p_i = \sum_{i=0}^n \gamma_i x_i$ Ex: $q=5$, $\begin{bmatrix} 1 \\ 3 \\ 2 \end{bmatrix} + 3 * \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 3 \end{bmatrix}$

- Typically for error correcting

- Reducing bits sent

- Encoding vector - coefficients
- Linear combination and vector sent
- Recover packets



[4]



Randomized Data Exchange (RDE) Algorithm

Iterate until all clients can recover all packets

- Select client c_{t_i} , the client who has the most information at iteration i
- Create encoding vector $\gamma_i^j = \begin{cases} \text{random element of } \mathbb{F}_q & \text{if } c_{t_i} \text{ has } x_j \\ 0 & \text{otherwise} \end{cases}$
- Compute $p_i = \sum_{x_j \in X} \gamma_i^j x_j$
- Broadcast p_i and γ_i



Theorem

The algorithm computes, with probability at least $1 - \frac{kn}{q}$ an optimal solution for the data exchange problem, provided that the size q is larger than n .

(Sprintson et al. 2010)

(Given k clients, n packets, and finite field of size q)




Motivation of Proof of Optimality

- Iteration i : Select client c_{t_i} with most information
- This client has information that *some* clients need -- C'
- $\mathbb{P}(\gamma_i \text{ *doesn't* increase rank of some client } c_j \in C')$
- $\mathbb{P}(< \gamma_i, \zeta_j \geq 0)$
- ζ_j -- contains information that client c_j *needs*


$$\langle \gamma_i, \zeta_j \rangle = 0$$

$$\sum_{k=0}^m \gamma_k \zeta_k = 0$$

$$\gamma_0 \zeta_0 + \sum_{k=1}^m \gamma_k \zeta_k = 0$$

$$\gamma_0 = - \frac{\sum_{k=1}^m \gamma_k \zeta_k}{\zeta_0}$$


Chosen i.i.d uniformly from a finite field of size q



Motivation of Proof of Optimality

- Iteration i : Select client c_{t_i} with most information
- This client has information that *some* clients need -- C'
- $\mathbb{P}(\gamma_i \text{ *doesn't* increase rank of some client } c_j \in C') = \frac{1}{q}$
- $\mathbb{P}(\gamma_i \text{ *doesn't* increase rank of *all* clients } c \in C') \leq \frac{k}{q}$
- $\mathbb{P}(\gamma_i \text{ *does* increase rank of all clients } c \in C') \geq 1 - \frac{k}{q}$

$$\left(1 - \frac{k}{q}\right)^{OPT} \geq \left(1 - \frac{k}{q}\right)^n \geq 1 - \frac{kn}{q}$$



Description of Experiments

- Bit counting
- Naïve algorithm
 - Optimizations
 - Counts size of packets that would be sent
- RDE algorithm
 - Counts size of linear combinations and encoding vectors
- Run types
 - has_one, 25%, 50%, 75%, missing_one
 - Random 10%, ..., random 90%
- Input sizes
- Field Size



Sample of results

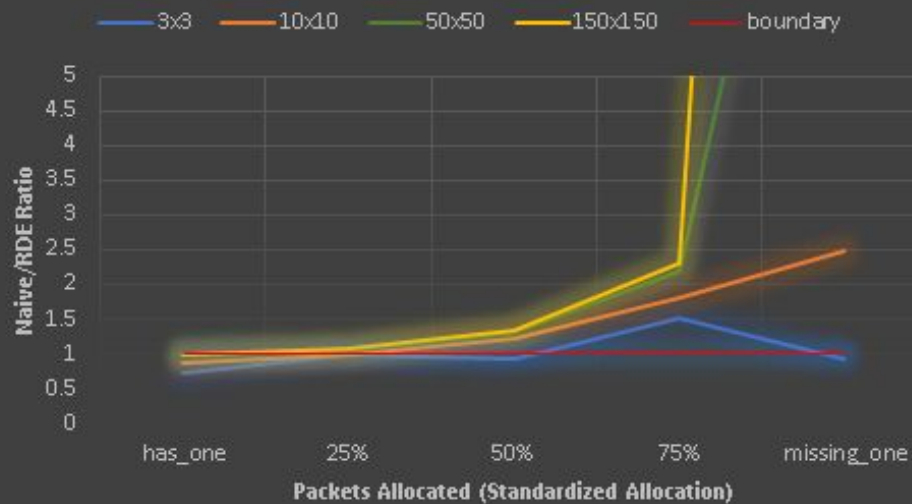
- Calculated average over 10 runs of RDE
- Ratio
 - (Naïve bits) / (RDE bits)

input_smallmed_gen.txt	has_one
562	636
562	651
562	654
562	634
562	655
562	635
562	630
562	635
562	635
562	625
	639

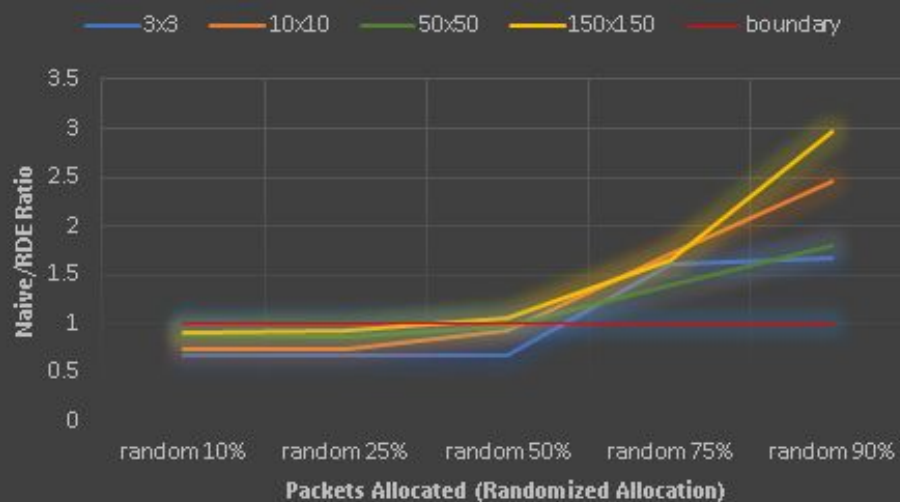
input_smallmed_gen.txt	missing_one
562	223
562	237
562	215
562	232
562	220
562	222
562	225
562	218
562	229
562	236
	225.7

Results

Normalized Difference vs. Packets Allocated to Clients

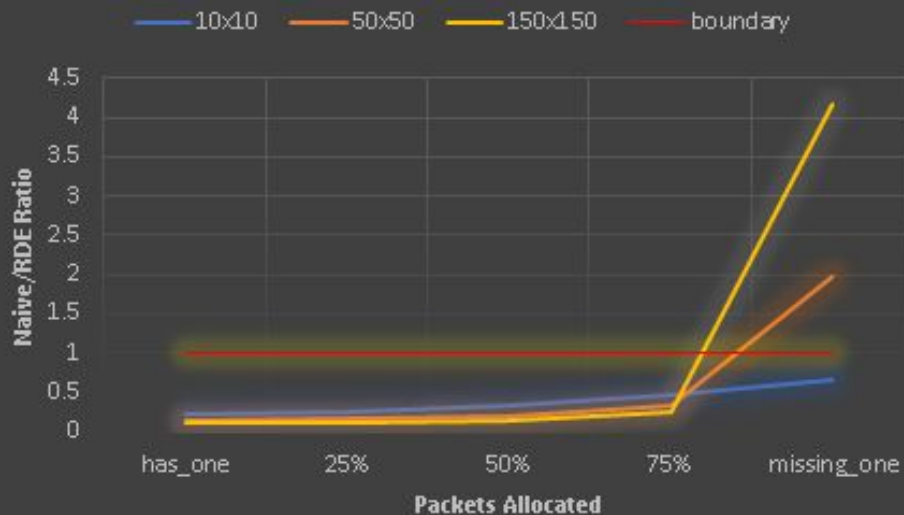


Normalized Difference vs. Packets Allocated to Clients

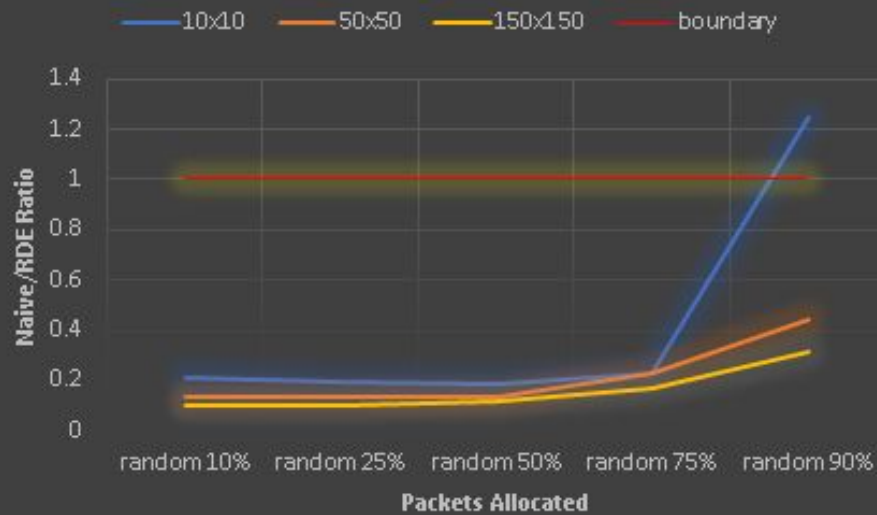


Results (Binary Codewords)

Bit Usage Ratio vs. Input Size (Standardized Packet Allocation)

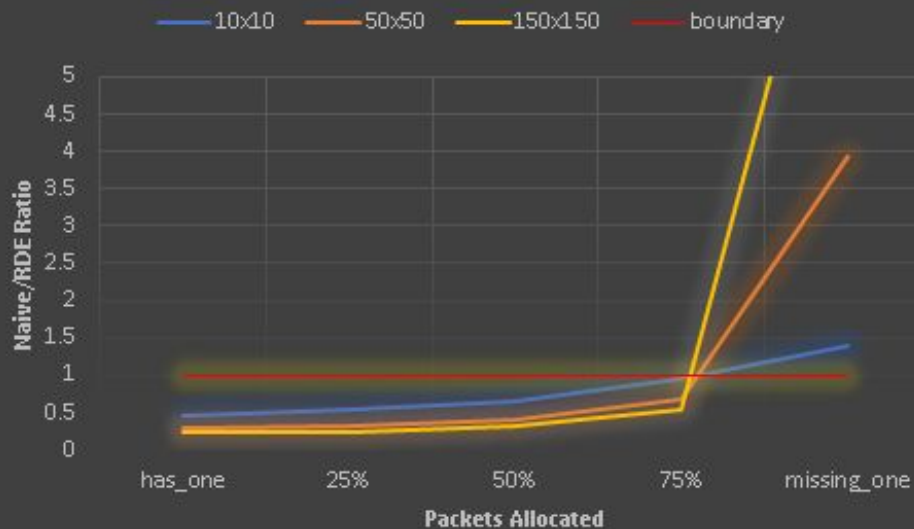


Bit Usage Ratio vs. Input Size (Randomized Packet Allocation)

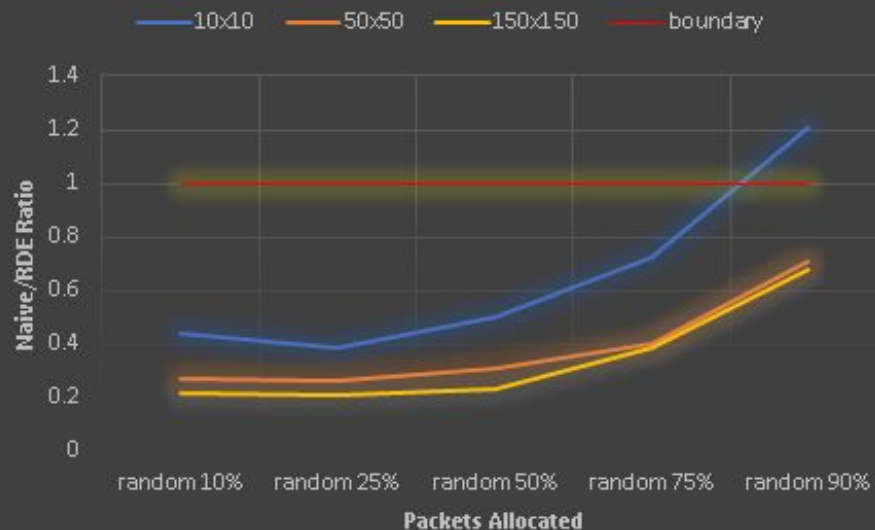


Results (Reduced Field Size)

Bit Usage Ratio vs. Input Size (Standardized Packet Allocation)



Bit Usage Ratio vs. Input Size (Randomized Packet Allocation)





Summary

- Algorithm finds an optimal solution with probability $1 - \frac{kn}{q}$
 - Guarantee certain probability of success
 - E.g. $q = 4kn \Rightarrow \frac{3}{4}$ prob. of success
- Multiple (simulated) iterations
- Linear coding is a useful technique in general



References

- [1] Sprintson, Alexander, et al. "A randomized algorithm and performance bounds for coded cooperative data exchange." *ISIT*. 2010.
- [2] <https://www.walmart.com/ip/Straight-Talk-Samsung-Galaxy-J7-Sky-Pro-16GB-LTE-No-Contract-Prepaid-SmartPhone-Black/840262752>
- [3] <https://www.vectorstock.com/royalty-free-vector/radio-base-station-linear-icon-concept-radio-base-vector-20917854>
- [4] <https://www.hindawi.com/journals/iidmb/2011/857847/fig3/>