Coded Acknowledgement with Random Subspaces

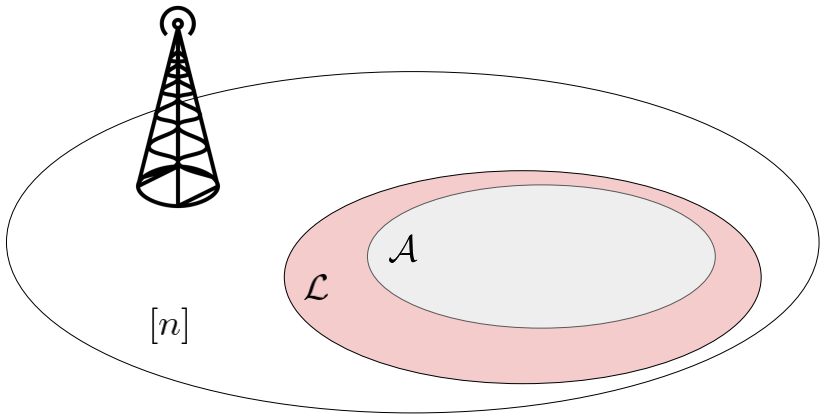
Nicholas Kwan*, Ryan Song†, Wei Yu*

*University of Toronto
†École Polytechnique Fédérale de Lausanne

June 30, 2026

Acknowledgement in Massive Random Access

- Communication scenario with $n \sim 10^4 - 10^6$ users and a central BS.
- Subset of ℓ users $\mathcal{L} \subset [n]$ contending for BS.
- BS detects some k users $\mathcal{A} \subset \mathcal{L}$.



Goal: communicate ACK to users in $\mathcal{A}$ and NACK to users in $\mathcal{L} \setminus \mathcal{A}$.

Naive Strategies

- Transmit list of users in $\mathcal{A}$.
  - k active users, each indexed by $\log(n)$-bit index, so this requires $k \log(n)$ bits.

Naive Strategies

- Transmit list of users in $\mathcal{A}$.
  - k active users, each indexed by $\log(n)$-bit index, so this requires $k \log(n)$ bits.
- This can be improved by noting that the order of the active users doesn't matter!
  - We can enumerate the k-subsets of $[n]$, and transmit index of active set.

Naive Strategies

- Transmit list of users in $\mathcal{A}$.
  - k active users, each indexed by $\log(n)$-bit index, so this requires $k \log(n)$ bits.
- This can be improved by noting that the order of the active users doesn't matter!
  - We can enumerate the k-subsets of $[n]$, and transmit index of active set.
  - There are $\binom{n}{k}$ such subsets, so this requires $\log \binom{n}{k} \gtrapprox k \log(n/k)$ bits.
  - Optimal for error-free acknowledgements!

Naive Strategies

- Transmit list of users in $\mathcal{A}$.
  - k active users, each indexed by $\log(n)$-bit index, so this requires $k \log(n)$ bits.
- This can be improved by noting that the order of the active users doesn't matter!
  - We can enumerate the k-subsets of $[n]$, and transmit index of active set.
  - There are $\binom{n}{k}$ such subsets, so this requires $\log \binom{n}{k} \gtrapprox k \log(n/k)$ bits.
  - Optimal for error-free acknowledgements!
- For $n = 10^6$, $k = 10^2$:
  - Listing users requires ~ 1900 bits,
  - Encoding k-subsets requires ~ 1300 bits.

Naive Strategies

- Transmit list of users in $\mathcal{A}$.
  - k active users, each indexed by $\log(n)$-bit index, so this requires $k \log(n)$ bits.
- This can be improved by noting that the order of the active users doesn't matter!
  - We can enumerate the k-subsets of $[n]$, and transmit index of active set.
  - There are $\binom{n}{k}$ such subsets, so this requires $\log \binom{n}{k} \gtrapprox k \log(n/k)$ bits.
  - Optimal for error-free acknowledgements!
- For $n = 10^6$, $k = 10^2$:
  - Listing users requires ~ 1900 bits,
  - Encoding k-subsets requires ~ 1300 bits.
- Each active user receives 1 bit of information—so this is 13 times the actual amount of information received.
- Can we reduce this common message length? How?

Acknowledgement with Error

Kalør et al. [KKP22] show that the common message length can be decreased significantly by introducing some nonzero error probability.

- For the optimal error-free acknowledgement, each codeword corresponds uniquely to a possible active set.
- When errors are introduced, one codeword can be used for multiple active sets, so fewer codewords are needed!

Let $\mathcal{A} \in \binom{[n]}{k}$ be the active set. Assume that the BS and each user have access to some common randomness $z \in \mathcal{Z}$.

- BS encodes the set of active users $\mathcal{A}$ using

$$f : \binom{[n]}{k} \times \mathcal{Z} \longrightarrow \{0,1\}^R.$$

Let $\mathcal{A} \in \binom{[n]}{k}$ be the active set. Assume that the BS and each user have access to some common randomness $z \in \mathcal{Z}$.

- BS encodes the set of active users $\mathcal{A}$ using

$$f : \binom{[n]}{k} \times \mathcal{Z} \longrightarrow \{0,1\}^R.$$

- Each user $u \in [n]$ decodes using

$$g_u : \{0,1\}^R \times \mathcal{Z} \to \{\text{ACK},\ \text{NACK}\}.$$

Let $\mathcal{A} \in \binom{[n]}{k}$ be the active set. Assume that the BS and each user have access to some common randomness $z \in \mathcal{Z}$.

- BS encodes the set of active users $\mathcal{A}$ using

$$f : \binom{[n]}{k} \times \mathcal{Z} \longrightarrow \{0,1\}^R.$$

- Each user $u \in [n]$ decodes using

$$g_u : \{0,1\}^R \times \mathcal{Z} \to \{\text{ACK},\ \text{NACK}\}.$$

- Missed detection error must be bounded:

$$\Pr\{g_u(f(\mathcal{A}, z), z) = \text{ACK}\} \geq 1 - \lambda_1 \quad \text{for all } u \in \mathcal{A}.$$

- False alarm error must be bounded:

$$\Pr\{g_u(f(\mathcal{A}, z), z) = \text{NACK}\} \geq 1 - \lambda_2 \quad \text{for all } u \notin \mathcal{A}.$$

Let $\mathcal{A} \in \binom{[n]}{k}$ be the active set. Assume that the BS and each user have access to some common randomness $z \in \mathcal{Z}$.

- BS encodes the set of active users $\mathcal{A}$ using

$$f : \binom{[n]}{k} \times \mathcal{Z} \longrightarrow \{0,1\}^R.$$

- Each user $u \in [n]$ decodes using

$$g_u : \{0,1\}^R \times \mathcal{Z} \to \{\text{ACK},\ \text{NACK}\}.$$

- Missed detection error must be bounded:

$$\Pr\{g_u(f(\mathcal{A}, z), z) = \text{ACK}\} \geq 1 - \lambda_1 \quad \text{for all } u \in \mathcal{A}.$$

- False alarm error must be bounded:

$$\Pr\{g_u(f(\mathcal{A}, z), z) = \text{NACK}\} \geq 1 - \lambda_2 \quad \text{for all } u \notin \mathcal{A}.$$

- Let $R^*(n, k, \lambda_1, \lambda_2)$ denote length of optimal coded acknowledgement scheme.
- Can we achieve $R^*(n, k, \lambda_1, \lambda_2)$ efficiently?

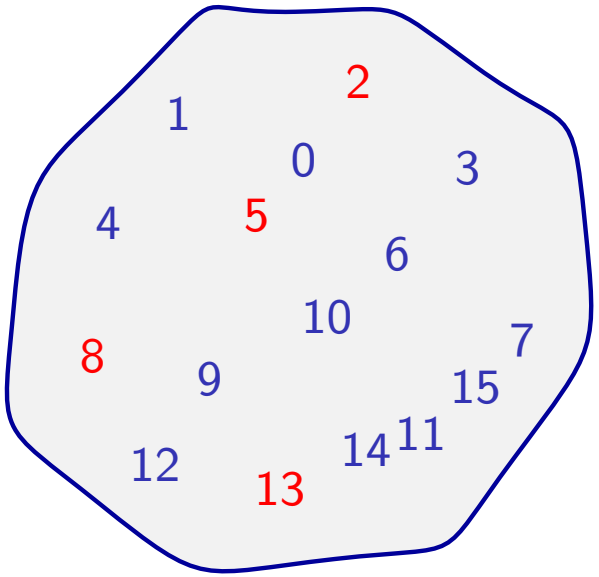
Porat's Scheme

Kalør et al. [KKP22] also introduce a scheme for $\lambda_1 = \lambda_2 = \lambda$ following [Por09] based on solving k linear equations over $\mathbb{F}_q$.

Porat's Scheme

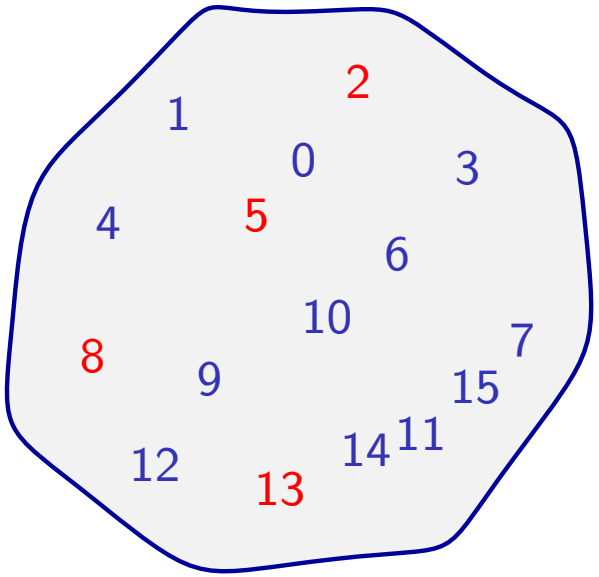
Kalør et al. [KKP22] also introduce a scheme for $\lambda_1 = \lambda_2 = \lambda$ following [Por09] based on solving k linear equations over $\mathbb{F}_q$.

Each user is assigned a random linear equation in k variables over $\mathbb{F}_q$. The set of active users defines a system of k equations in k variables.



Kalør et al. [KKP22] also introduce a scheme for $\lambda_1 = \lambda_2 = \lambda$ following [Por09] based on solving k linear equations over $\mathbb{F}_q$.

Each user is assigned a random linear equation in k variables over $\mathbb{F}_q$. The set of active users defines a system of k equations in k variables.



Ex: $n = 16$, $k = 4$, $q = 5$.

$$\begin{array}{lccccccccc}
\text{User 2:} & x_1 & + & x_2 & + & 3x_3 & + & 2x_4 & = & 0 \\
\text{User 5:} & x_1 & + & x_2 & + & x_3 & + & 2x_4 & = & 3 \\
\text{User 8:} & 3x_1 & + & 2x_2 & + & 4x_3 & + & x_4 & = & 0 \\
\text{User 13:} & 4x_1 & + & 0x_2 & + & 0x_3 & + & 3x_4 & = & 3
\end{array}$$

Solve this system of linear equations in $\mathbb{F}_q$ and transmit the solution (x_1, x_2, x_3, x_4).

- Solve the system of equations by performing Gaussian elimination over $\mathbb{F}_q$:

$$\left[\begin{array}{cccc|c} 1 & 1 & 3 & 2 & 0 \\ 1 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 & 0 \\ 4 & 0 & 0 & 3 & 3 \end{array}\right] \xrightarrow{RREF} \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1.$$

- A user decodes to ACK if the transmitted solution satisfies its equation.
  $\Pr(x_i\text{'s satisfy a random equation}) = 1/q$.

- Solve the system of equations by performing Gaussian elimination over $\mathbb{F}_q$:

$$\left[\begin{array}{cccc|c} 1 & 1 & 3 & 2 & 0 \\ 1 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 & 0 \\ 4 & 0 & 0 & 3 & 3 \end{array}\right] \xrightarrow{RREF} \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1.$$

- A user decodes to ACK if the transmitted solution satisfies its equation. $\Pr(x_i\text{'s satisfy a random equation}) = 1/q$.
- If no solution exists, give NACK to all users. $\Pr(\text{no solution exists}) \lesssim 1/q$.

- Solve the system of equations by performing Gaussian elimination over $\mathbb{F}_q$:

$$\left[\begin{array}{cccc|c} 1 & 1 & 3 & 2 & 0 \\ 1 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 & 0 \\ 4 & 0 & 0 & 3 & 3 \end{array}\right] \xrightarrow{RREF} \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1.$$

- A user decodes to ACK if the transmitted solution satisfies its equation. $\Pr(x_i\text{'s satisfy a random equation}) = 1/q$.
- If no solution exists, give NACK to all users. $\Pr(\text{no solution exists}) \lesssim 1/q$.
- By choosing $q \geq 1/\lambda$, we have $\Pr(\text{missed detection}) \lesssim \Pr(\text{false alarm}) \leq \lambda$.

- Solve the system of equations by performing Gaussian elimination over $\mathbb{F}_q$:

$$\left[\begin{array}{cccc|c} 1 & 1 & 3 & 2 & 0 \\ 1 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 & 0 \\ 4 & 0 & 0 & 3 & 3 \end{array}\right] \xrightarrow{RREF} \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1.$$

- A user decodes to ACK if the transmitted solution satisfies its equation. $\Pr(x_i\text{'s satisfy a random equation}) = 1/q$.
- If no solution exists, give NACK to all users. $\Pr(\text{no solution exists}) \lesssim 1/q$.
- By choosing $q \geq 1/\lambda$, we have $\Pr(\text{missed detection}) \lesssim \Pr(\text{false alarm}) \leq \lambda$.
- Expressing $x_i \in \mathbb{F}_q$ takes $\lceil \log(q) \rceil$ bits, so the message length is $k\lceil \log(1/\lambda) \rceil$.

- Solve the system of equations by performing Gaussian elimination over $\mathbb{F}_q$:

$$\left[\begin{array}{cccc|c} 1 & 1 & 3 & 2 & 0 \\ 1 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 & 0 \\ 4 & 0 & 0 & 3 & 3 \end{array}\right] \xrightarrow{RREF} \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1.$$

- A user decodes to ACK if the transmitted solution satisfies its equation. $\Pr(x_i\text{'s satisfy a random equation}) = 1/q$.
- If no solution exists, give NACK to all users. $\Pr(\text{no solution exists}) \lesssim 1/q$.
- By choosing $q \geq 1/\lambda$, we have $\Pr(\text{missed detection}) \lesssim \Pr(\text{false alarm}) \leq \lambda$.
- Expressing $x_i \in \mathbb{F}_q$ takes $\lceil \log(q) \rceil$ bits, so the message length is $k\lceil \log(1/\lambda) \rceil$.
- For $n = 10^6$, $k = 100$, $\lambda_1 = \lambda_2 = 10^{-2}$, the common message length can be reduced to 700 bits from naive scheme of 1900 bits.

- Solve the system of equations by performing Gaussian elimination over $\mathbb{F}_q$:

$$\left[\begin{array}{cccc|c} 1 & 1 & 3 & 2 & 0 \\ 1 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 & 0 \\ 4 & 0 & 0 & 3 & 3 \end{array}\right] \xrightarrow{RREF} \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1.$$

- A user decodes to ACK if the transmitted solution satisfies its equation. $\Pr(x_i\text{'s satisfy a random equation}) = 1/q$.
- If no solution exists, give NACK to all users. $\Pr(\text{no solution exists}) \lesssim 1/q$.
- By choosing $q \geq 1/\lambda$, we have $\Pr(\text{missed detection}) \lesssim \Pr(\text{false alarm}) \leq \lambda$.
- Expressing $x_i \in \mathbb{F}_q$ takes $\lceil \log(q) \rceil$ bits, so the message length is $k\lceil \log(1/\lambda) \rceil$.
- For $n = 10^6$, $k = 100$, $\lambda_1 = \lambda_2 = 10^{-2}$, the common message length can be reduced to 700 bits from naive scheme of 1900 bits.
- However, the complexity of Gaussian elimination in $\mathbb{F}_q$ is $O(k^3 \log(1/\lambda))$.

- Solve the system of equations by performing Gaussian elimination over $\mathbb{F}_q$:

$$\left[\begin{array}{cccc|c} 1 & 1 & 3 & 2 & 0 \\ 1 & 1 & 1 & 2 & 3 \\ 3 & 2 & 4 & 1 & 0 \\ 4 & 0 & 0 & 3 & 3 \end{array}\right] \xrightarrow{RREF} \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array}\right]$$

$$x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 1.$$

- A user decodes to ACK if the transmitted solution satisfies its equation. $\Pr(x_i\text{'s satisfy a random equation}) = 1/q$.
- If no solution exists, give NACK to all users. $\Pr(\text{no solution exists}) \lesssim 1/q$.
- By choosing $q \geq 1/\lambda$, we have $\Pr(\text{missed detection}) \lesssim \Pr(\text{false alarm}) \leq \lambda$.
- Expressing $x_i \in \mathbb{F}_q$ takes $\lceil \log(q) \rceil$ bits, so the message length is $k\lceil \log(1/\lambda) \rceil$.
- For $n = 10^6$, $k = 100$, $\lambda_1 = \lambda_2 = 10^{-2}$, the common message length can be reduced to 700 bits from naive scheme of 1900 bits.
- However, the complexity of Gaussian elimination in $\mathbb{F}_q$ is $O(k^3 \log(1/\lambda))$.

This complexity increases as λ gets small! Can we decrease it?

Main Results

We give a novel coding scheme based on losslessly compressing random binary subspaces:

We give a novel coding scheme based on losslessly compressing random binary subspaces:

- By compressing the subspace naively, we can perform acknowledgement with $O(k^3)$ bit complexity and common message length

$$\lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil.$$

We give a novel coding scheme based on losslessly compressing random binary subspaces:

- By compressing the subspace naively, we can perform acknowledgement with $O(k^3)$ bit complexity and common message length

$$\lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil.$$

- By compressing the subspace optimally, we obtain the following achievable common message length:

$$R^*(n,k,\lambda_1,\lambda_2) \leq \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2.$$

We give a novel coding scheme based on losslessly compressing random binary subspaces:

- By compressing the subspace naively, we can perform acknowledgement with $O(k^3)$ bit complexity and common message length

$$\lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil.$$

- By compressing the subspace optimally, we obtain the following achievable common message length:

$$R^*(n,k,\lambda_1,\lambda_2) \leq \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2.$$

- Converse shows achievability tight up to $O(k)$ for large n:

$$R^*(n,k,\lambda_1,\lambda_2) \geq (1-\lambda_1)k \log\left(\frac{1}{\lambda_2 + \frac{(1-\lambda_1)k}{n}}\right) - O(k).$$

Basic idea:

1. Assign each user u a random binary vector $h(u) \in \mathbb{F}_2^d$ of some length d.

Basic idea:

1. Assign each user u a random binary vector $h(u) \in \mathbb{F}_2^d$ of some length d.
2. Transmit the subspace generated by the active users, i.e., $W = \mathrm{span}(h(\mathcal{A}))$.

Basic idea:

1. Assign each user u a random binary vector $h(u) \in \mathbb{F}_2^d$ of some length d.
2. Transmit the subspace generated by the active users, i.e., $W = \mathrm{span}(h(\mathcal{A}))$.

$\iff$ Losslessly compress the unique RREF generator matrix G' that generates W.

Basic idea:

1. Assign each user u a random binary vector $h(u) \in \mathbb{F}_2^d$ of some length d.
2. Transmit the subspace generated by the active users, i.e., $W = \mathrm{span}(h(\mathcal{A}))$.

   $\iff$ Losslessly compress the unique RREF generator matrix G' that generates W.
3. User u checks whether or not its $h(u) \in W$.
   - If $h(u) \in W$, it decodes to ACK.
   - Otherwise, it decodes to NACK.

Basic idea:

1. Assign each user u a random binary vector $h(u) \in \mathbb{F}_2^d$ of some length d.
2. Transmit the subspace generated by the active users, i.e., $W = \mathrm{span}(h(\mathcal{A}))$.

   $\iff$ Losslessly compress the unique RREF generator matrix G' that generates W.
3. User u checks whether or not its $h(u) \in W$.
   - If $h(u) \in W$, it decodes to ACK.
   - Otherwise, it decodes to NACK.

   $\iff$ Each user forms a parity check matrix H from G' and computes $h(u)H^T$.

Example:

Suppose $k = 3$ and we choose $d = 5$. Suppose the active users a_1, a_2, a_3 have been given the vectors

$$h(a_1) = [1\ 1\ 0\ 1\ 1],$$
$$h(a_2) = [0\ 0\ 1\ 0\ 0],$$
$$h(a_3) = [0\ 0\ 1\ 0\ 1].$$

Example:

Suppose $k = 3$ and we choose $d = 5$. Suppose the active users a_1, a_2, a_3 have been given the vectors

$$h(a_1) = [1\ 1\ 0\ 1\ 1],$$
$$h(a_2) = [0\ 0\ 1\ 0\ 0],$$
$$h(a_3) = [0\ 0\ 1\ 0\ 1].$$

Then forming the generator matrix and row-reducing yields

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \xrightarrow{RREF} G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

G' is compressed losslessly then transmitted over a noiseless channel to the users.

Each listening user forms the parity-check matrix H upon recovering G':

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \longrightarrow H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Each listening user forms the parity-check matrix H upon recovering G':

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \longrightarrow H = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Listening users decode by multiplying their binary vector with H:

$$[1\ 1\ 0\ 1\ 1] \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = [0\ 0] \quad \longrightarrow \quad \text{ACK}$$

$$[1\ 1\ 0\ 0\ 1] \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = [0\ 1] \quad \longrightarrow \quad \text{NACK}$$

We will make some tweaks to our basic coding strategy:

- As it stands, this scheme will always achieve $\lambda_1 = 0$, since G' is a generator matrix of a code containing $h(u)$ for each $u \in \mathcal{A}$.
- To get $\lambda_1 > 0$ probability of missed detection, choose a random subset of $k' = \lceil (1 - \lambda_1)k \rceil$ active users $\mathcal{A}' \subset \mathcal{A}$ to acknowledge.

We will make some tweaks to our basic coding strategy:

- As it stands, this scheme will always achieve $\lambda_1 = 0$, since G' is a generator matrix of a code containing $h(u)$ for each $u \in \mathcal{A}$.
- To get $\lambda_1 > 0$ probability of missed detection, choose a random subset of $k' = \lceil (1 - \lambda_1)k \rceil$ active users $\mathcal{A}' \subset \mathcal{A}$ to acknowledge.
- Ensure that G' has rank k' by adding linearly independent rows as needed.

Now we choose d as follows:

$$d = k' + \lceil \log(1/\lambda_2) \rceil.$$

k': Dimension of the code generated by G' to satisfy missed detection error.

Now we choose d as follows:

$$d = k' + \lceil \log(1/\lambda_2) \rceil.$$

k': Dimension of the code generated by G' to satisfy missed detection error.

$\lceil \log(1/\lambda_2) \rceil$: Makes d the smallest integer such that false alarm probability is at most λ_2.

Now we choose d as follows:

$$d = k' + \lceil \log(1/\lambda_2) \rceil.$$

k': Dimension of the code generated by G' to satisfy missed detection error.

$\lceil \log(1/\lambda_2) \rceil$: Makes d the smallest integer such that false alarm probability is at most λ_2.

Indeed, since G' is the generator matrix of a code W of dimension k', then for any $u \notin \mathcal{A}$, $h(u)$ is independent of W so that

$$\Pr\{\text{u decodes to ACK}\} = \frac{|W|}{2^d} = \frac{2^{k'}}{2^{k' + \lceil \log(1/\lambda_2) \rceil}} = 2^{\lceil \log(1/\lambda_2) \rceil} \leq \lambda_2.$$

- The common message encodes a $k' \times d$ full rank binary matrix in RREF, where $k' = \lceil (1-\lambda_1)k \rceil$. Thus, we have

$$R^*(n, k, \lambda_1, \lambda_2) \leq \log\Big(\#\text{ full rank } k' \times d \text{ binary matrices in RREF}\Big).$$

- The common message encodes a $k' \times d$ full rank binary matrix in RREF, where $k' = \lceil (1-\lambda_1)k \rceil$. Thus, we have

$$R^*(n, k, \lambda_1, \lambda_2) \leq \log\Big(\#\text{ full rank } k' \times d \text{ binary matrices in RREF}\Big).$$

- Let $\begin{bmatrix} d \\ k' \end{bmatrix}_2$ be the number of full rank $k' \times d$ binary matrices in RREF.

- The common message encodes a $k' \times d$ full rank binary matrix in RREF, where $k' = \lceil (1-\lambda_1)k \rceil$. Thus, we have

$$R^*(n, k, \lambda_1, \lambda_2) \leq \log\Big(\#\text{ full rank } k' \times d \text{ binary matrices in RREF}\Big).$$

- Let $\begin{bmatrix} d \\ k' \end{bmatrix}_2$ be the number of full rank $k' \times d$ binary matrices in RREF.
- A lower bound for $\begin{bmatrix} d \\ k' \end{bmatrix}_2$ is given by the number of RREF matrices whose pivot columns are the first k' columns:

$$\left[\begin{array}{c|c} I_{k'} & P \end{array}\right] \Big\} k'$$
$$\underbrace{\phantom{I_{k'}}}_{k'} \quad \underbrace{}_{d-k'}$$

Since there are $k'(d-k')$ entries in P, there are $2^{k'(d-k')}$ such matrices, and thus $2^{k'(d-k)} \leq \begin{bmatrix} d \\ k' \end{bmatrix}_2$.

- For any other choice of pivot columns, there are fewer "free" entries of P, since some must be fixed to 0.
- The lower bound actually gives tight scaling, and the following bounds are known:

$$2^{k'(d-k')} \leq \begin{bmatrix} d \\ k' \end{bmatrix}_2 \leq 4 * 2^{k'(d-k')}.$$

- For any other choice of pivot columns, there are fewer "free" entries of P, since some must be fixed to 0.
- The lower bound actually gives tight scaling, and the following bounds are known:

$$2^{k'(d-k')} \leq \begin{bmatrix} d \\ k' \end{bmatrix}_2 \leq 4 * 2^{k'(d-k')}.$$

- So taking the logarithm gives

$$R^*(n, k, \lambda_1, \lambda_2) \leq k'(d - k') + 2 = \lceil (1 - \lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2.$$

- For any other choice of pivot columns, there are fewer "free" entries of P, since some must be fixed to 0.
- The lower bound actually gives tight scaling, and the following bounds are known:

$$2^{k'(d-k')} \leq \begin{bmatrix} d \\ k' \end{bmatrix}_2 \leq 4 * 2^{k'(d-k')}.$$

- So taking the logarithm gives

$$R^*(n, k, \lambda_1, \lambda_2) \leq k'(d - k') + 2 = \lceil (1 - \lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2.$$

- For $n = 10^6$, $k = 100$, $\lambda_1 = \lambda_2 = 10^{-2}$, the common message length can in theory be reduced to 695 bits from naive scheme of 1900 bits.

Practical Coding Scheme

A naive way to compress G' is to specify the k' pivot columns using an indicator vector v_{piv}, and the entries in the nonpivot columns G'_{np}.

A naive way to compress G' is to specify the k' pivot columns using an indicator vector v_{piv}, and the entries in the nonpivot columns G'_{np}.

Example:

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The pivot columns are 1, 3, and 5. The nonpivot columns are red.

$$v_{piv} = \underbrace{[1\ 0\ 1\ 0\ 1]}_{d = k' + \lceil \log(1/\lambda_2) \rceil \text{ bits}}, \quad G'_{np} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{k'(d-k') = k' \lceil \log(1/\lambda_2) \rceil \text{ bits}}.$$

A naive way to compress G' is to specify the k' pivot columns using an indicator vector v_{piv}, and the entries in the nonpivot columns G'_{np}.

Example:

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The pivot columns are 1, 3, and 5. The nonpivot columns are red.

$$v_{piv} = \underbrace{[1\ 0\ 1\ 0\ 1]}_{d = k' + \lceil \log(1/\lambda_2) \rceil \text{ bits}}, \quad G'_{np} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{k'(d-k') = k' \lceil \log(1/\lambda_2) \rceil \text{ bits}}.$$

Transmit v_{piv} using $k' + \lceil \log(1/\lambda_2) \rceil$ bits, G'_{np} using $k' \lceil \log(1/\lambda_2) \rceil$ bits.

A naive way to compress G' is to specify the k' pivot columns using an indicator vector v_{piv}, and the entries in the nonpivot columns G'_{np}.

Example:

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The pivot columns are 1, 3, and 5. The nonpivot columns are red.

$$v_{piv} = \underbrace{[1\ 0\ 1\ 0\ 1]}_{d=k'+\lceil \log(1/\lambda_2) \rceil \text{ bits}}, \quad G'_{np} = \underbrace{\begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{k'(d-k')=k'\lceil \log(1/\lambda_2) \rceil \text{ bits}}.$$

Transmit v_{piv} using $k' + \lceil \log(1/\lambda_2) \rceil$ bits, G'_{np} using $k' \lceil \log(1/\lambda_2) \rceil$ bits.

For $n = 10^6$, $k = 100$, $\lambda_1 = \lambda_2 = 10^{-2}$, the common message length can be reduced to 799 bits from naive scheme of 1900 bits.

Theorem

Consider a massive random access scenario in which BS wishes to acknowledge k users out of a total of n users with per-user missed detection and false alarm error probabilities at most λ_1 and λ_2, respectively. The minimum fixed-length common message $R^*(n, k, \lambda_1, \lambda_2)$ is bounded from above as

$$R^* \leq \lceil (1-\lambda_1)k \rceil \lceil \log\left(1/\lambda_2\right) \rceil + 2.$$

The practical parity-check scheme has bit complexity $O(k^3)$ and achieves

$$R = \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil.$$

Theorem

Consider a massive random access scenario in which BS wishes to acknowledge k users out of a total of n users with per-user missed detection and false alarm error probabilities at most λ_1 and λ_2, respectively. The minimum fixed-length common message $R^*(n, k, \lambda_1, \lambda_2)$ is bounded from above as

$$R^* \leq \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2.$$

The practical parity-check scheme has bit complexity $O(k^3)$ and achieves

$$R = \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil.$$

- Encoding complexity is that of row-reducing a $k' \times (k' + \lceil \log(1/\lambda_2) \rceil)$ matrix in $\mathbb{F}_2$.

Theorem

Consider a massive random access scenario in which BS wishes to acknowledge k users out of a total of n users with per-user missed detection and false alarm error probabilities at most λ_1 and λ_2, respectively. The minimum fixed-length common message $R^*(n, k, \lambda_1, \lambda_2)$ is bounded from above as

$$R^* \leq \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2.$$

The practical parity-check scheme has bit complexity $O(k^3)$ and achieves

$$R = \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil.$$

- Encoding complexity is that of row-reducing a $k' \times (k' + \lceil \log(1/\lambda_2) \rceil)$ matrix in $\mathbb{F}_2$.
- Arithmetic in $\mathbb{F}_2$ is $\lceil \log(1/\lambda_2) \rceil$-times more efficient than in $\mathbb{F}_q$ for $q \geq 1/\lambda_2$.

Theorem

Consider a massive random access scenario in which BS wishes to acknowledge k users out of a total of n users with per-user missed detection and false alarm error probabilities at most λ_1 and λ_2, respectively. The minimum fixed-length common message $R^*(n, k, \lambda_1, \lambda_2)$ is bounded from above as

$$R^* \leq \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + 2.$$

The practical parity-check scheme has bit complexity $O(k^3)$ and achieves

$$R = \lceil (1-\lambda_1)k \rceil \lceil \log(1/\lambda_2) \rceil + \lceil (1-\lambda_1)k \rceil + \lceil \log(1/\lambda_2) \rceil.$$

- Encoding complexity is that of row-reducing a $k' \times (k' + \lceil \log(1/\lambda_2) \rceil)$ matrix in $\mathbb{F}_2$.
- Arithmetic in $\mathbb{F}_2$ is $\lceil \log(1/\lambda_2) \rceil$-times more efficient than in $\mathbb{F}_q$ for $q \geq 1/\lambda_2$.
- The parity-check scheme can be seen as a dual of the scheme in [KKP22]:
  - Rather than finding a solution to a random system of equations, we find a system of equations admitting a random set of solutions.
  - Such a system of equations always exists!

Converse

- Let A be a random variable denoting set of active users,
- Let $\hat{A}$ denote the set of users who believe they have been acknowledged, i.e.,

$$\hat{A} = \{u \in [n] : g_u(f(A, Z), Z) = \text{ACK}\}.$$

Converse

- Let A be a random variable denoting set of active users,
- Let $\hat{A}$ denote the set of users who believe they have been acknowledged, i.e.,

$$\hat{A} = \{u \in [n] : g_u(f(A, Z), Z) = \text{ACK}\}.$$

- Conditioned on common randomness Z, $A - f(A, Z) - \hat{A}$ is a Markov chain. Therefore:

$$\begin{aligned}
H(f(A,Z)) &\geq H(f(A,Z)|Z) \\
&\geq I(f(A,Z); \hat{A}|Z) \\
&\geq I(A; \hat{A}|Z) && \text{(Conditional data processing ineq.)} \\
&= H(A|Z) - H(A|\hat{A}, Z) \\
&= H(A) - H(A|\hat{A}, Z) && (A \text{ and } Z \text{ are independent}) \\
&\geq I(A; \hat{A}).
\end{aligned}$$

Converse

- Since A is distributed uniformly, $H(A) = \log \binom{n}{k}$.

Converse

- Since A is distributed uniformly, $H(A) = \log \binom{n}{k}$.
- $H(A|\hat{A})$ is roughly log of number of sets A such that given $\hat{A}$ satisfies (λ_1, λ_2) error probabilities.

Converse

- Since A is distributed uniformly, $H(A) = \log \binom{n}{k}$.
- $H(A|\hat{A})$ is roughly log of number of sets A such that given $\hat{A}$ satisfies (λ_1, λ_2) error probabilities.
- Log-concavity of binomial coefficient lets us bound $H(A|\hat{A})$ by conditioning on number of missed detections and applying Jensen's inequality.

- Since A is distributed uniformly, $H(A) = \log \binom{n}{k}$.
- $H(A|\hat{A})$ is roughly log of number of sets A such that given $\hat{A}$ satisfies (λ_1, λ_2) error probabilities.
- Log-concavity of binomial coefficient lets us bound $H(A|\hat{A})$ by conditioning on number of missed detections and applying Jensen's inequality.
- Therefore:

$$R^*(n, k, \lambda_1, \lambda_2) \geq (1-\lambda_1)k \log\left(\frac{1}{\lambda_2 + \frac{(1-\lambda_1)k}{n}}\right) \underbrace{-k(1+2\log(e)) - \log(\lambda_1 k + 1) - \log(e).}_{\text{losses from Jensen's inequality, approximating binomial coefficient}}$$

[AD89]:

"*Let $S_1, \ldots, S_n$ be sailors on a ship, and let sailor S_i be associated with lady L_i.*
In a stormy night one sailor, say S_j, drowns in the ocean. One could now broadcast his name to the radio stations of the country from which all sailors are known to come, hoping that the lady L_j listens to the news, so that she hears about the tragic event.
However, this takes $\lceil \log_2 n \rceil$ bits and the news is (primarily) of interest only to one lady.

[AD89]:

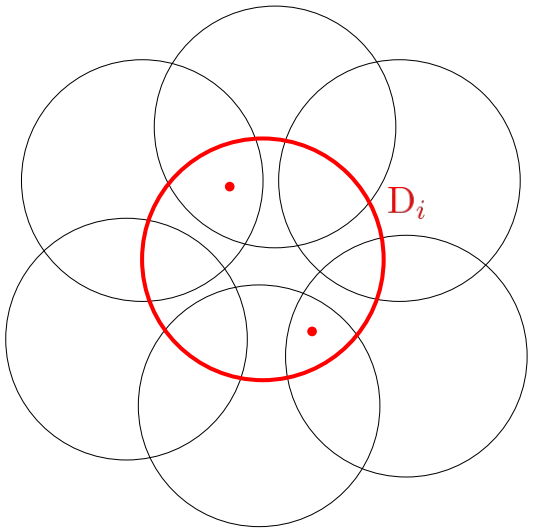
"*Let $S_1, \ldots, S_n$ be sailors on a ship, and let sailor S_i be associated with lady L_i.*
In a stormy night one sailor, say S_j, drowns in the ocean. One could now broadcast his name to the radio stations of the country from which all sailors are known to come, hoping that the lady L_j listens to the news, so that she hears about the tragic event.
However, this takes $\lceil \log_2 n \rceil$ bits and the news is (primarily) of interest only to one lady.
If we now permit a certain error probability, which is not much of a price in an imperfect (as the tragedy shows) world, then by our result $O(\log \log n)$ bits suffice!"

[AD89]:

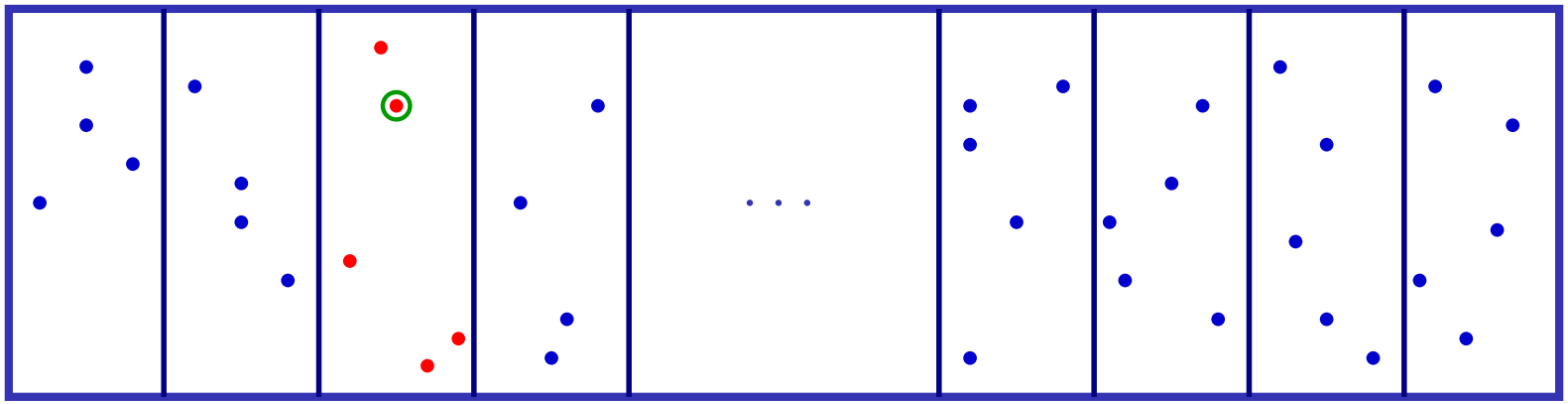
"*Let $S_1, \ldots, S_n$ be sailors on a ship, and let sailor S_i be associated with lady L_i.*
In a stormy night one sailor, say S_j, drowns in the ocean. One could now broadcast his name to the radio stations of the country from which all sailors are known to come, hoping that the lady L_j listens to the news, so that she hears about the tragic event.
However, this takes $\lceil \log_2 n \rceil$ bits and the news is (primarily) of interest only to one lady.
If we now permit a certain error probability, which is not much of a price in an imperfect (as the tragedy shows) world, then by our result $O(\log \log n)$ bits suffice!"

Identification code for the noiseless binary channel:

- Design $\mathcal{D}_i \subset \{0,1\}^m, \forall i \in [n]$, with $m = O(\log \log n)$.
- Identify user $i \in [n]$ by transmitting a randomly selected m-bit sequence in $\mathcal{D}_i$.
- Control false alarm by ensuring $\frac{|\mathcal{D}_i \cap \mathcal{D}_j|}{|\mathcal{D}_i|} \leq \lambda_2, \forall i, j$.
- Explicit constructions are very complicated!

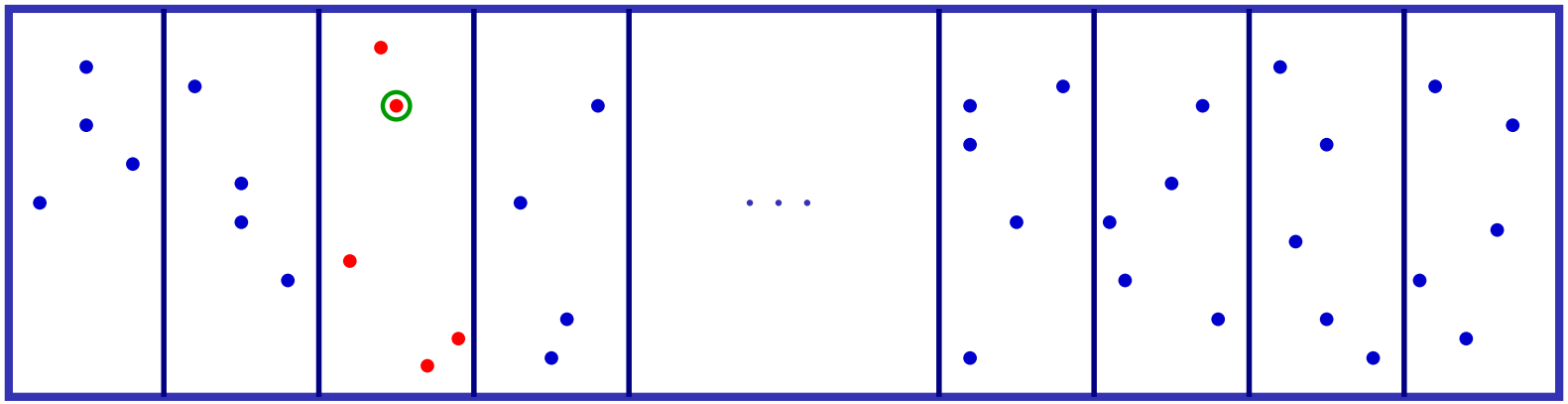


- Identification becomes trivial if common randomness is available:



1. Randomly hash the n users into $\lceil 1/\lambda_2 \rceil$ bins,
2. Transmit the index of the bin containing the active user.

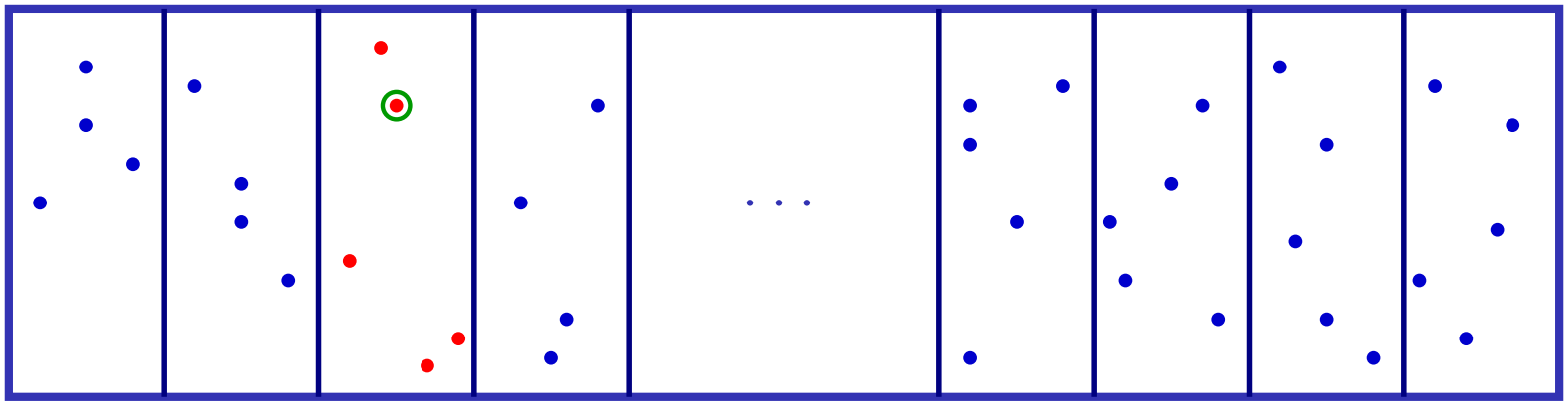
- Identification becomes trivial if common randomness is available:



1. Randomly hash the n users into $\lceil 1/\lambda_2 \rceil$ bins,
2. Transmit the index of the bin containing the active user.

- Message length is $\lceil \log(1/\lambda_2) \rceil$ bits. Error probability is λ_2.

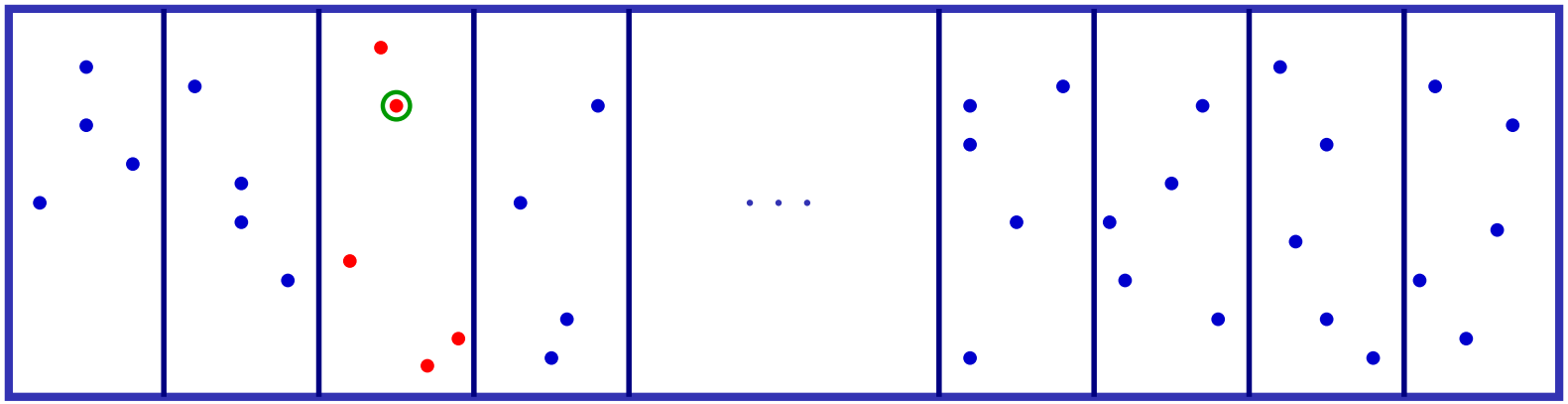
- Identification becomes trivial if common randomness is available:



1. Randomly hash the n users into $\lceil 1/\lambda_2 \rceil$ bins,
2. Transmit the index of the bin containing the active user.

- Message length is $\lceil \log(1/\lambda_2) \rceil$ bits. Error probability is λ_2.
- Recover scaling of [AD89] by taking $\lambda_2 = 1/\log(n)$, which vanishes as $n \to \infty$.

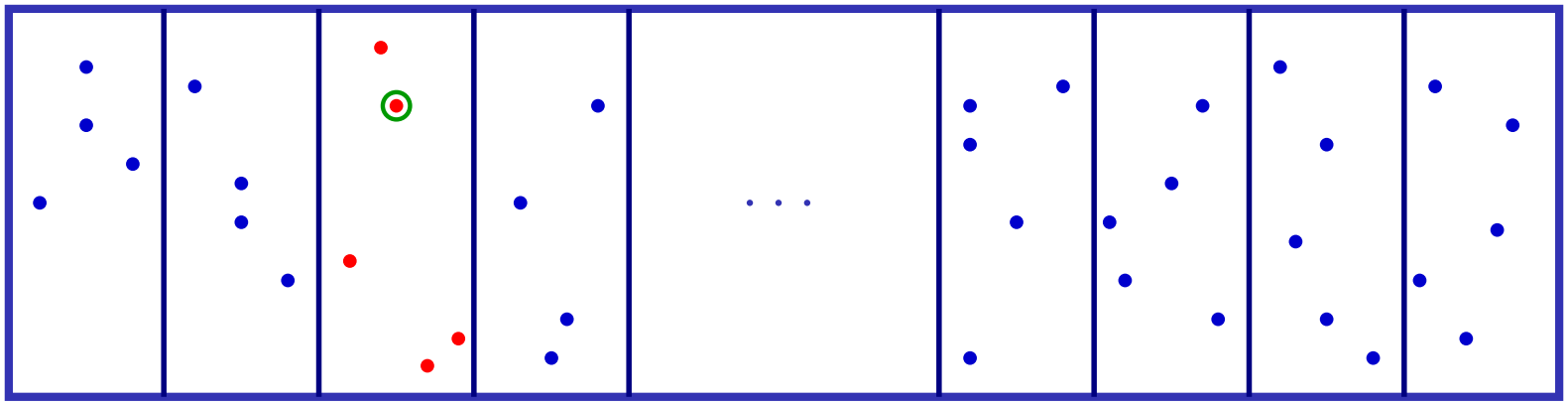
- Identification becomes trivial if common randomness is available:



1. Randomly hash the n users into $\lceil 1/\lambda_2 \rceil$ bins,
2. Transmit the index of the bin containing the active user.

- Message length is $\lceil \log(1/\lambda_2) \rceil$ bits. Error probability is λ_2.
- Recover scaling of [AD89] by taking $\lambda_2 = 1/\log(n)$, which vanishes as $n \to \infty$.
- We can even beat double exponential scaling, e.g., by letting $\lambda_2 = 1/\log\log n$.

- Identification becomes trivial if common randomness is available:



1. Randomly hash the n users into $\lceil 1/\lambda_2 \rceil$ bins,
2. Transmit the index of the bin containing the active user.

- Message length is $\lceil \log(1/\lambda_2) \rceil$ bits. Error probability is λ_2.
- Recover scaling of [AD89] by taking $\lambda_2 = 1/\log(n)$, which vanishes as $n \to \infty$.
- We can even beat double exponential scaling, e.g., by letting $\lambda_2 = 1/\log\log n$.
- Since transmitting $d = \lceil \log(1/\lambda_2) \rceil$ bits can be thought of as encoding a one-dimensional subspace in $\mathbb{F}_2^d$, this is exactly the earlier coding strategy!

- Introduced novel coding strategy by communicating subspace spanned by random binary vectors given to active users. This gives

$$R^*(n, k, \lambda_1, \lambda_2) \leq \lceil (1-\lambda_1)k \rceil \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil + 2.$$

- Introduced novel coding strategy by communicating subspace spanned by random binary vectors given to active users. This gives

$$R^*(n,k,\lambda_1,\lambda_2) \leq \lceil (1-\lambda_1)k \rceil \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil + 2.$$

- Converse result showing that achievability scaling is tight:

$$R^*(n,k,\lambda_1,\lambda_2) \geq (1-\lambda_1)k \log\left(\frac{1}{\lambda_2 + \frac{(1-\lambda_1)k}{n}}\right) - O(k).$$

Conclusion

- Introduced novel coding strategy by communicating subspace spanned by random binary vectors given to active users. This gives

$$R^*(n, k, \lambda_1, \lambda_2) \leq \lceil (1-\lambda_1)k \rceil \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil + 2.$$

- Converse result showing that achievability scaling is tight:

$$R^*(n, k, \lambda_1, \lambda_2) \geq (1-\lambda_1)k \log\left(\frac{1}{\lambda_2 + \frac{(1-\lambda_1)k}{n}}\right) - O(k).$$

- Produced a practical algorithm based on naive compression of RREF generator matrix of subspace spanned by active set.
  - $\log(1/\lambda_2)$ reduction in encoding complexity compared to previous scheme.
  - Leading term matches above upper bound.

Conclusion

- Introduced novel coding strategy by communicating subspace spanned by random binary vectors given to active users. This gives

$$R^*(n,k,\lambda_1,\lambda_2) \leq \lceil (1-\lambda_1)k \rceil \left\lceil \log\left(\frac{1}{\lambda_2}\right) \right\rceil + 2.$$

- Converse result showing that achievability scaling is tight:

$$R^*(n,k,\lambda_1,\lambda_2) \geq (1-\lambda_1)k \log\left(\frac{1}{\lambda_2 + \frac{(1-\lambda_1)k}{n}}\right) - O(k).$$

- Produced a practical algorithm based on naive compression of RREF generator matrix of subspace spanned by active set.
  - $\log(1/\lambda_2)$ reduction in encoding complexity compared to previous scheme.
  - Leading term matches above upper bound.
- Results can be extended to transmission of bits after acknowledgement.

Thanks for listening!

References I


R. Ahlswede and G. Dueck.
Identification via channels.
IEEE Trans. Inf. Theory, 35(1):15–29, 1989.


Anders E. Kalør, Radosław Kotaba, and Petar Popovski.
Common message acknowledgments: Massive ARQ protocols for wireless access.
IEEE Trans. Commun., 70(8):5258–5270, 2022.


Ely Porat.
An optimal bloom filter replacement based on matrix solving.
In Fourth Int. Comp. Sci. Symp. Russia, pages 263–273. Springer, 2009.