Common Message Acknowledgments: Massive ARQ Protocols for Wireless Access

Algorithmic Structures for Uncoordinated Communications and Statistical Inference in Exceedingly Large Spaces

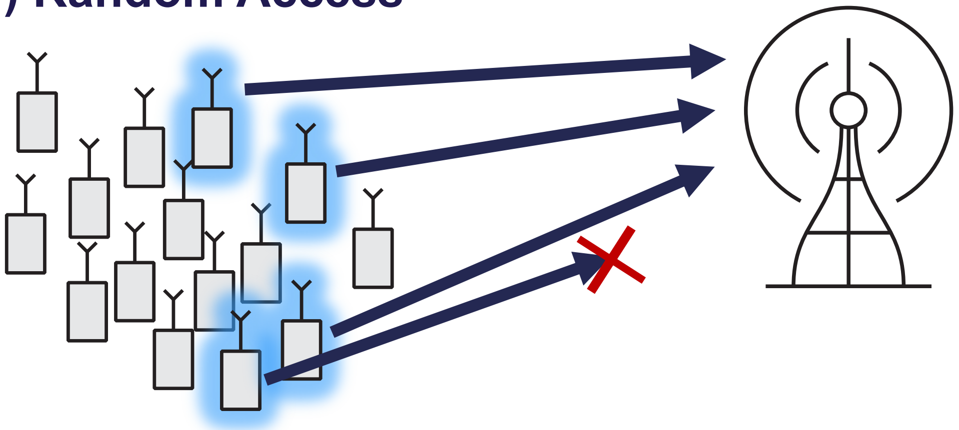
March 2024

Anders E. Kalør (aek@es.aau.dk)

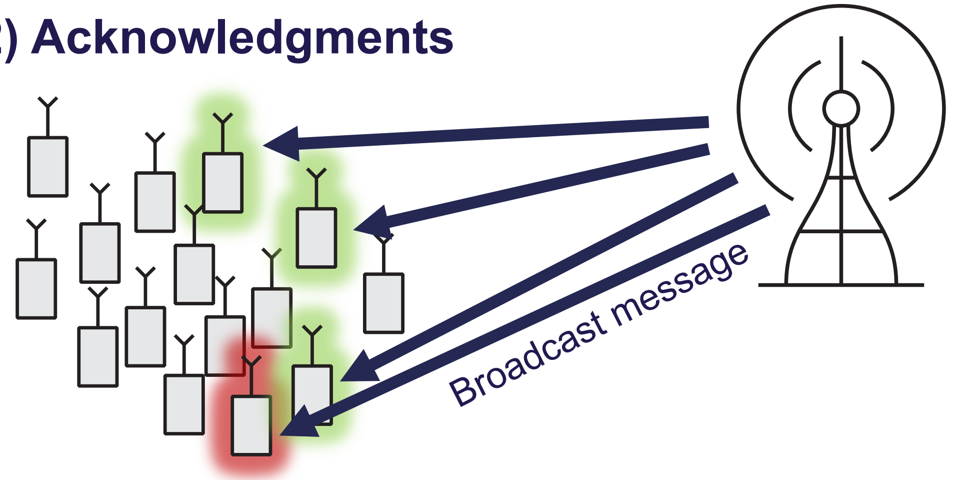
AALBORG UNIVERSITY
DENMARK

Common Acknowledgments

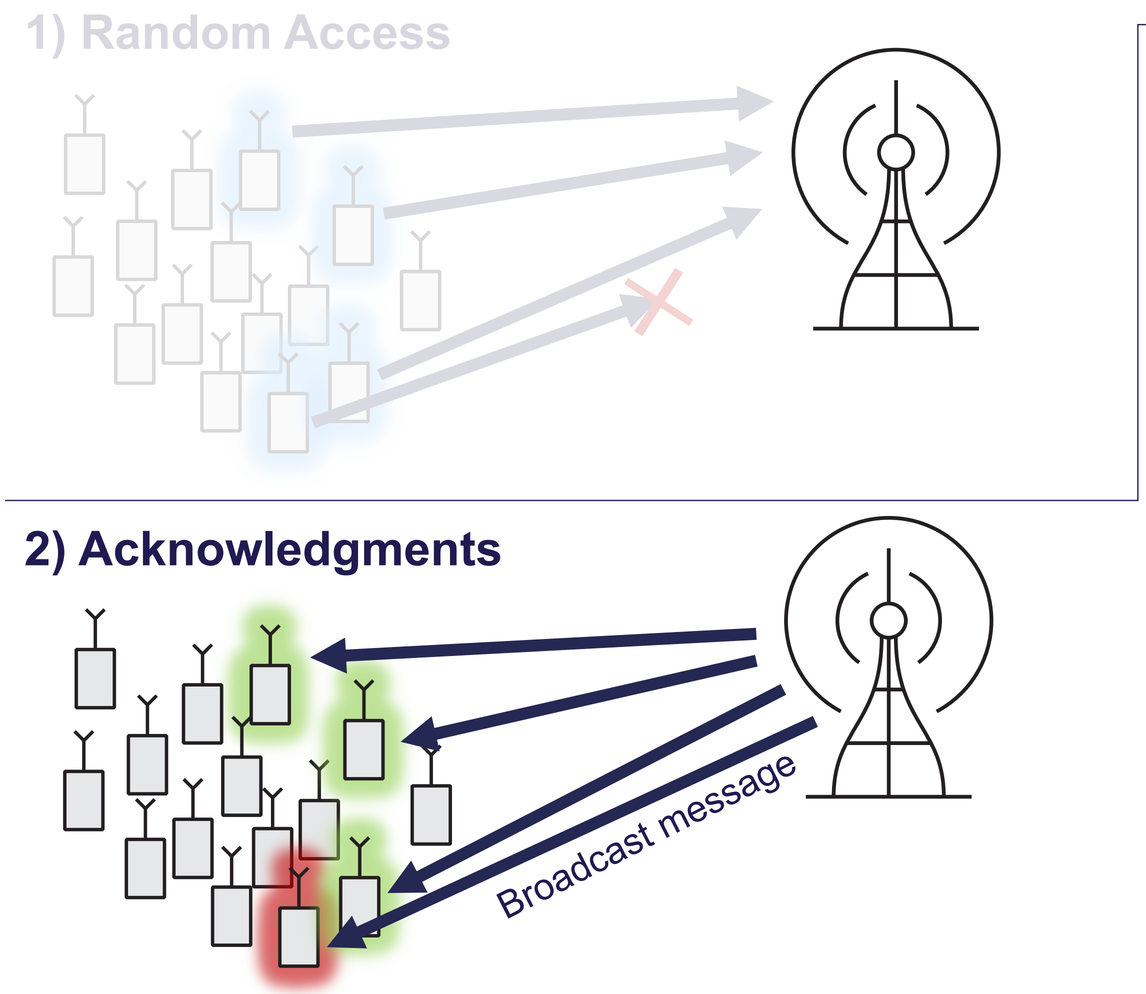
1) Random Access



2) Acknowledgments



Common Acknowledgments



Naive solution: **concatenation**

user 65	user 103	user 211

Can we do better?

What are the limits and trade-offs?

A. E. Kalør, R. Kotaba and P. Popovski, "**Common Message Acknowledgments: Massive ARQ Protocols for Wireless Access**," in *IEEE Transactions on Communications*, vol. 70, no. 8, pp. 5258-5270, Aug. 2022.

Outline

Part 1
Information Theoretic Bounds

Part 2
Practical Schemes

Part 3
Applications in ARQ protocols

Outline

Part 1
Information Theoretic Bounds

Part 2
Practical Schemes

Part 3
Applications in ARQ protocols

Formal Problem Definition

- $[N] = \{1,2,\ldots,N\}$: set of potentially active users (e.g., $N = 2^{32}$)
- $\mathcal{S} = \{s_1, s_2, \ldots, s_K\} \sim \mathcal{U}\left(\binom{[N]}{K}\right)$: set of K ***recovered*** users
- $K \ll N$, assumed to be constant (e.g., $K = 100$)

ACK encoder

$$f\colon \binom{[N]}{K} \to \underbrace{\{0,1\}^B}_{\text{ACK message length}}$$

User n's ACK decoder

$$g_n\colon \{0,1\}^B \to \{\mathrm{ACK}, \mathrm{NACK}\}$$

Error Types

False **positives** (false alarms)

$$\varepsilon_{\mathrm{fp}} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{P}\Big(g_n\big(f(\mathcal{S})\big) = \mathrm{ACK} \,\Big|\, n \notin \mathcal{S}\Big)$$

False **negatives** (missed detections)

$$\varepsilon_{\mathrm{fn}} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{P}\Big(g_n\big(f(\mathcal{S})\big) = \mathrm{NACK} \,\Big|\, n \in \mathcal{S}\Big)$$

Error-free Encoding

$\varepsilon_{\mathrm{fp}} = \varepsilon_{\mathrm{fn}} = 0$

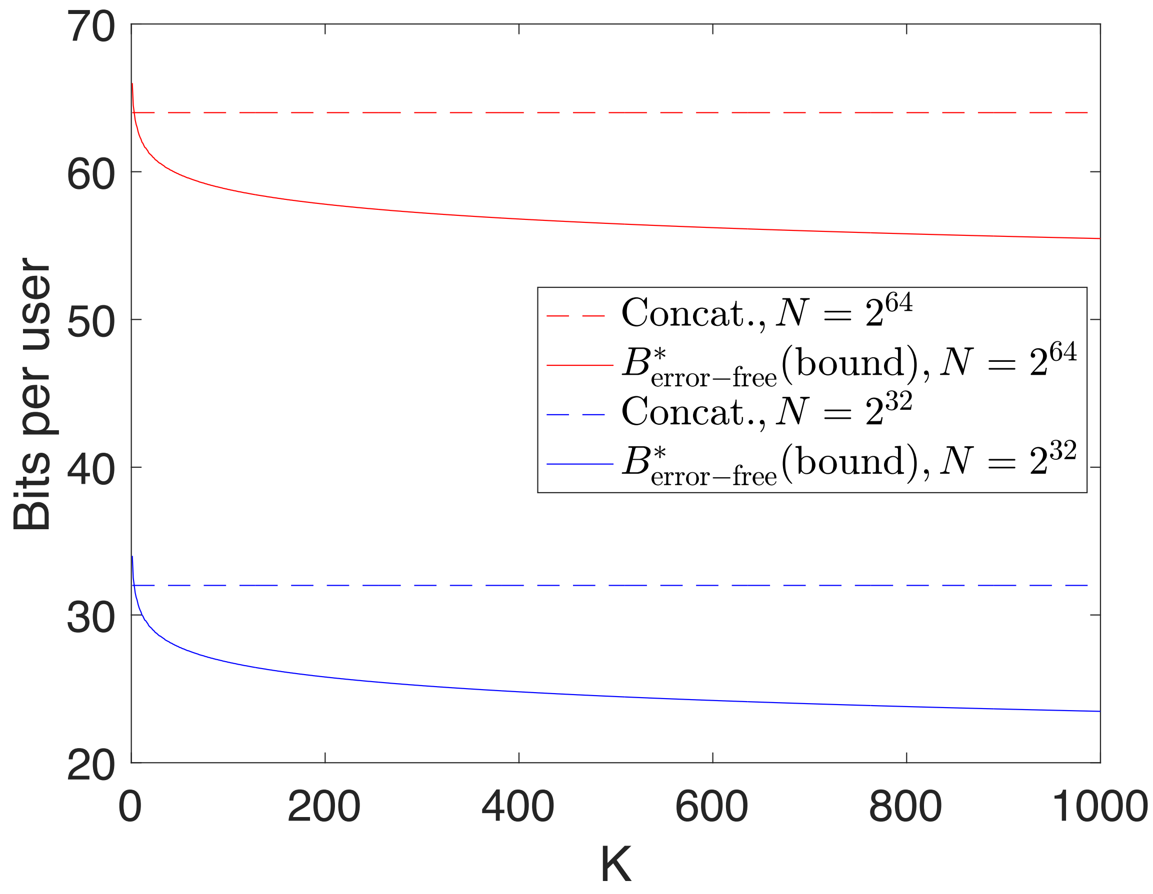
There are $\binom{N}{K}$ ways to pick the K recovered users, so we need

$$B^*_{\mathrm{error-free}} = \left\lceil \log_2 \binom{N}{K} \right\rceil \quad \text{[bits]}$$

$$\geq \left\lceil \mathrm{K} \log_2 \left(\frac{N}{K}\right) \right\rceil \quad \text{[bits]}$$

Error-free Encoding

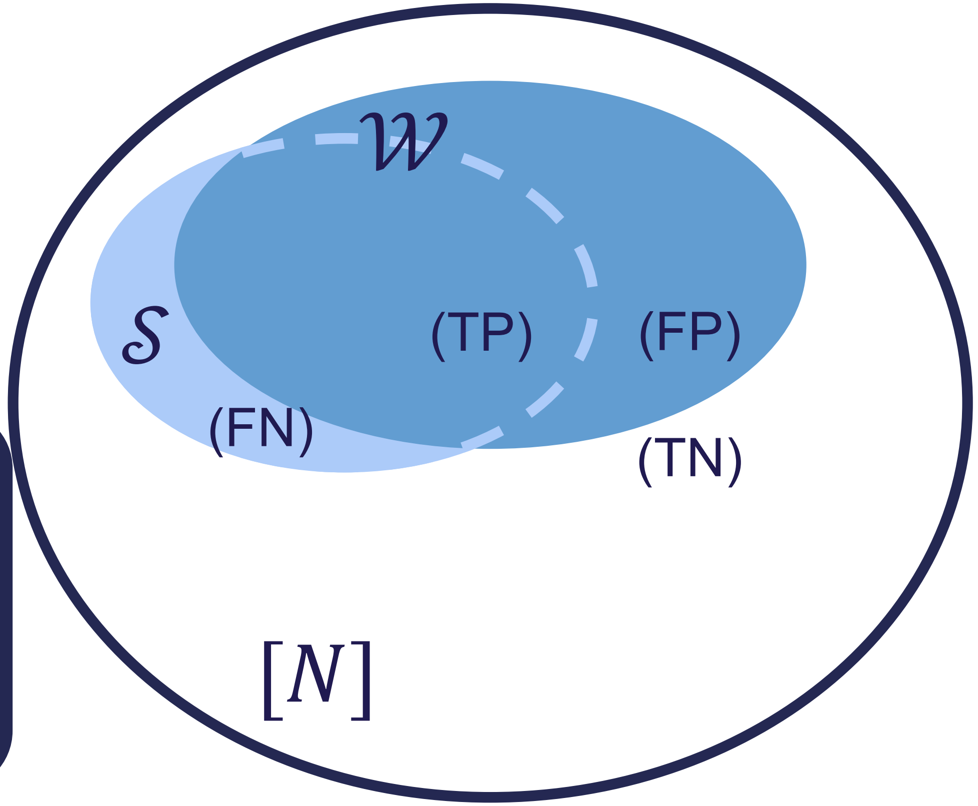
Note that $B^*_{\text{error-free}} = \left\lceil \log_2 \binom{N}{K} \right\rceil \leq \left\lceil K \log_2 \left(\frac{Ne}{K} \right) \right\rceil$ bits



Encoding with Errors

$\varepsilon_{\text{fp}} > 0, \qquad \varepsilon_{\text{fn}} \geq 0$

Each ACK message $\mathcal{W}$ can be used for several sets of recovered users $\mathcal{S}$



$$B^*_{\text{fp,fn}} \geq K \log_2\left(\frac{1}{\varepsilon_{\text{fp}} + \frac{K}{N}}\right) - K \log_2\left(\frac{e}{1-\varepsilon_{\text{fn}}}\right) - \varepsilon_{fn} K \log_2\left(\frac{1-\varepsilon_{\text{fn}}}{\varepsilon_{\text{fn}}\left(\varepsilon_{\text{fp}} + \frac{K}{N}\right)}\right) - \log_2 K \text{ [bits]}$$

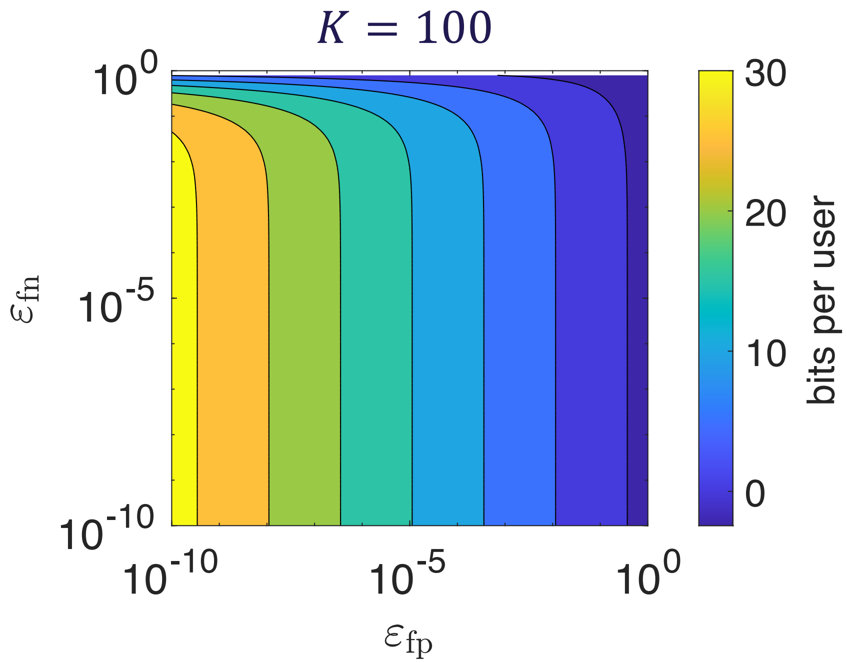
R. Pagh and F. F. Rodler, "Lossy dictionaries," in *Eur. Symp. Algorithms*. Springer, 2001, pp. 300–311

Encoding with Errors

Does not depend on N as $N \to \infty$ for fixed K

$$B^*_{\text{fp,fn}} \geq K \log_2\left(\frac{1}{\varepsilon_{\text{fp}} + \cancel{\frac{K}{N}}}\right) - K \log_2\left(\frac{e}{1-\varepsilon_{\text{fn}}}\right) - \varepsilon_{\text{fn}} K \log_2\left(\frac{1-\varepsilon_{\text{fn}}}{\varepsilon_{\text{fn}}\left(\varepsilon_{\text{fp}} + \cancel{\frac{K}{N}}\right)}\right) - \log_2 K$$

False positives give the highest gains

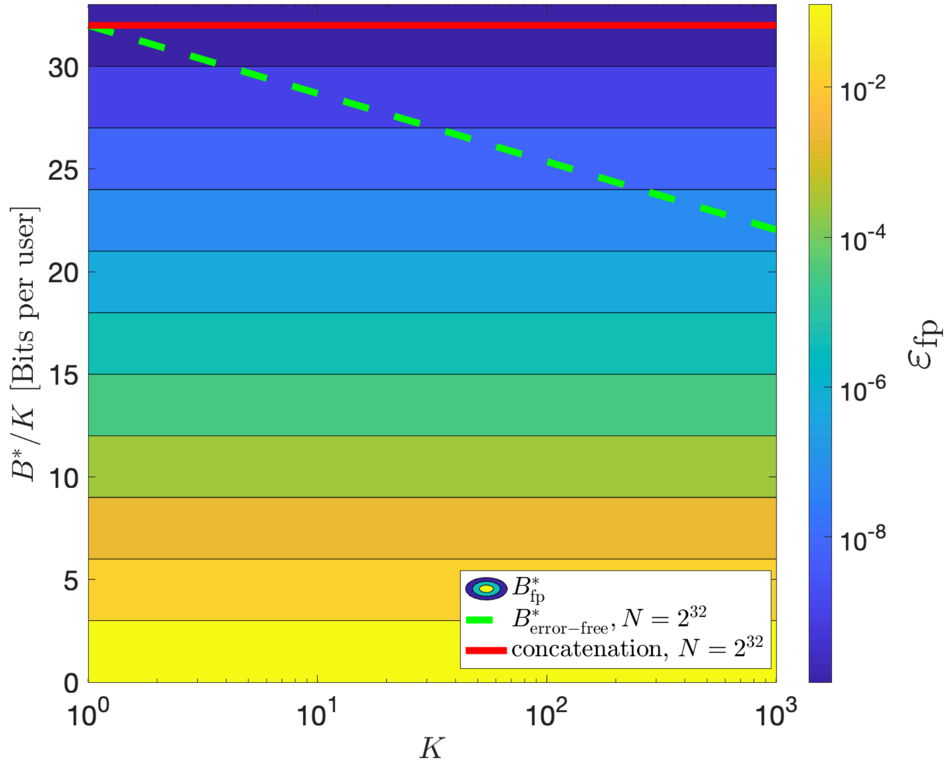


Encoding with Errors, $\varepsilon_{\text{fn}} = 0$

$\varepsilon_{\text{fp}} > 0, \qquad \varepsilon_{\text{fn}} = 0$

For large N:

$$B^*_{\text{fp}} = K \log_2\left(\frac{1}{\varepsilon_{\text{fp}}}\right) \pm \mathcal{O}(\log\log N)$$



L. Carter, et al., "Exact and approximate membership testers," in Proc. Tenth annu. ACM Symp. Theory Comp. (STOC). ACM Press, 1978.
M. Dietzfelbinger and R. Pagh, "Succinct data structures for retrieval and approximate membership," in Int. Colloq. Automata, Languages, and Program. Springer, 2008, pp. 385–396.

Outline

Part 1
Information Theoretic Bounds

Part 2
Practical Schemes

Part 3
Applications in ARQ protocols

Bloom Filter

Encoding:

$$\mathcal{S} = \{s_1, s_2, s_3\}$$

X																		
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bloom Filter

Encoding:

$$\mathcal{S} = \{s_1, s_2, s_3\}$$

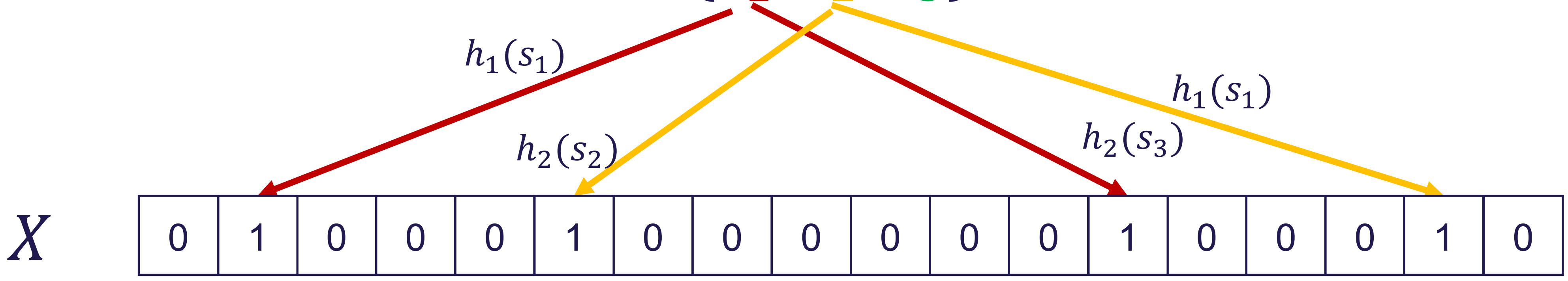
$h_1(s_1)$ $h_2(s_3)$

X	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bloom Filter

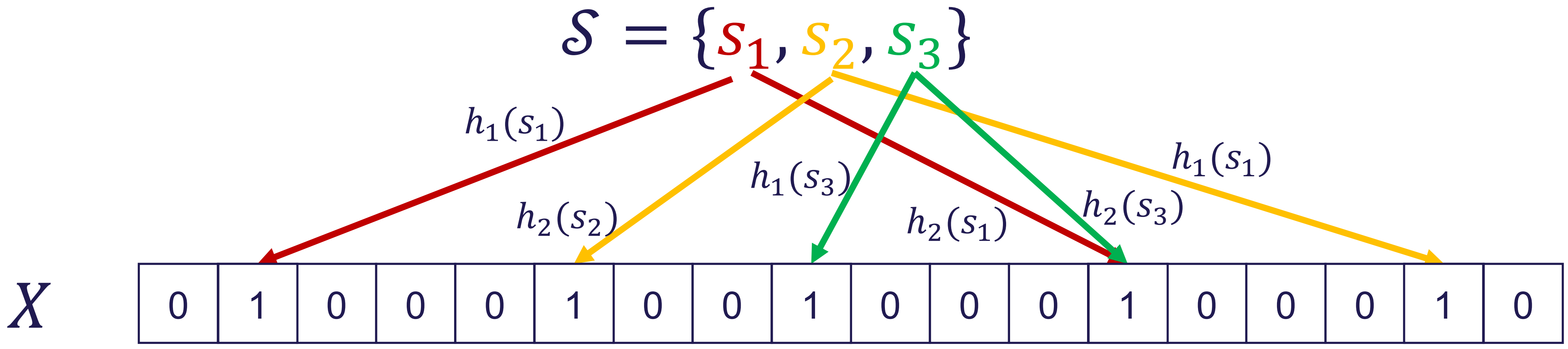
Encoding:

$$\mathcal{S} = \{s_1, s_2, s_3\}$$



Bloom Filter

Encoding:

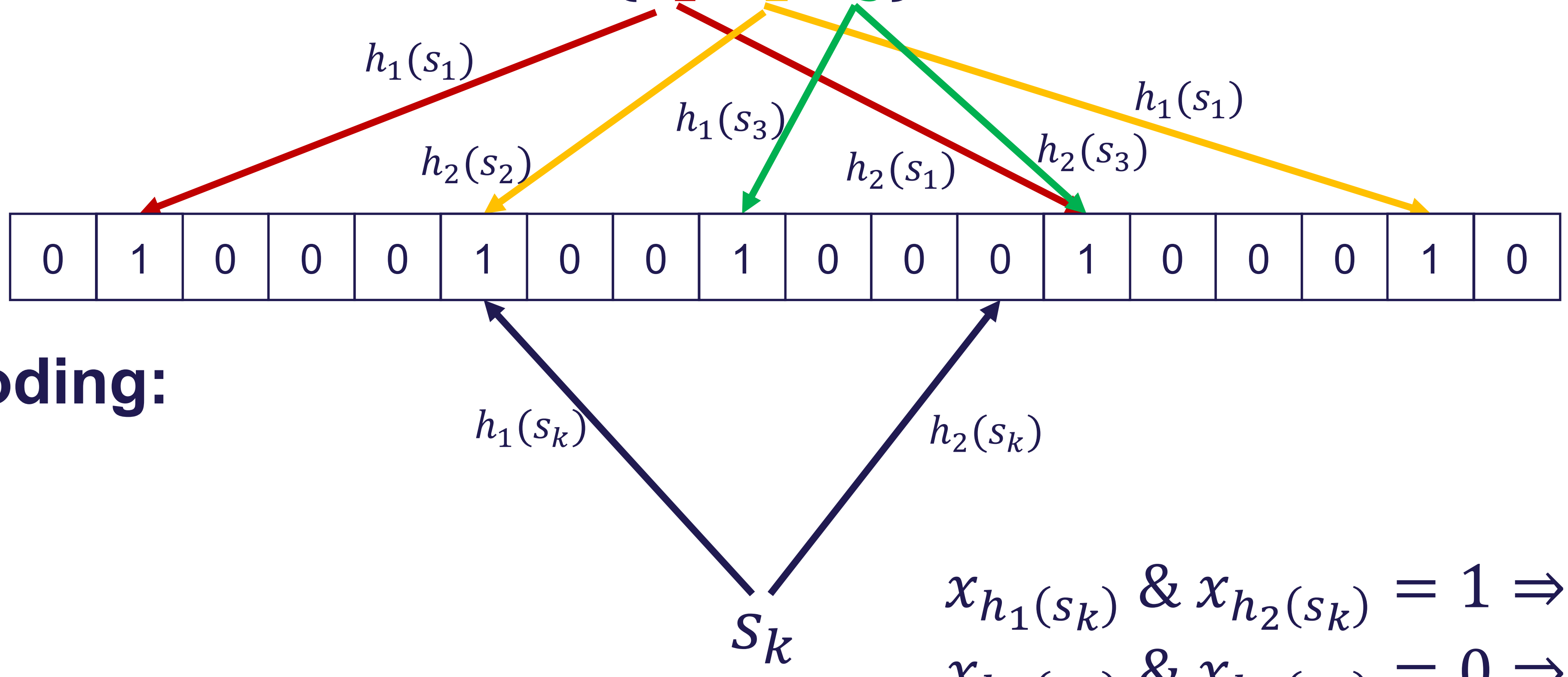


Bloom Filter

Encoding:

$$\mathcal{S} = \{s_1, s_2, s_3\}$$

X



Decoding:

$$x_{h_1(s_k)} \,\&\, x_{h_2(s_k)} = 1 \Rightarrow \text{ACK}$$

$$x_{h_1(s_k)} \,\&\, x_{h_2(s_k)} = 0 \Rightarrow \text{NACK}$$

Bloom Filter Analysis

After optimizing the number of hash functions and the message length it can be shown that

$$B_{\mathrm{bf}} = K \log_2(e) \log_2\left(\frac{1}{\varepsilon_{\mathrm{fp}}}\right)$$

A factor $\log_2(e) \approx 1.44$ larger than the asymptotic bound

A. Broder and M. Mitzenmacher, “Network applications of bloom filters: A survey,” Internet Math., vol. 1, no. 4, p. 485–509, Jan 2004.

Linear Equations

Consider the set of K linear equations constructed using hashes of the user ids

$$\mathcal{S} = \{s_1, s_2, s_3\}$$

$$\begin{bmatrix} h_1^{(1)}(s_1) & h_1^{(2)}(s_1) & h_1^{(3)}(s_1) \\ h_1^{(1)}(s_2) & h_1^{(2)}(s_2) & h_1^{(3)}(s_2) \\ h_1^{(1)}(s_3) & h_1^{(2)}(s_3) & h_1^{(3)}(s_3) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} h_2(s_1) \\ h_2(s_2) \\ h_2(s_3) \end{bmatrix}$$

unknown vector

All hash functions are
$[N] \to \mathrm{GF}(2^p)$

M. Dietzfelbinger and R. Pagh, "Succinct data structures for retrieval and approximate membership," in *Int. Colloq. Automata, Languages, and Program.* Springer, 2008, pp. 385–396.
E. Porat, "An optimal bloom filter replacement based on matrix solving," in Int. Comput. Sci. Symp. Russia. Springer, 2009, pp. 263–273.

Linear Equations

$$\begin{bmatrix} h_1^{(1)}(s_1) & h_1^{(2)}(s_1) & h_1^{(3)}(s_1) \\ h_1^{(1)}(s_2) & h_1^{(2)}(s_2) & h_1^{(3)}(s_2) \\ h_1^{(1)}(s_3) & h_1^{(2)}(s_3) & h_1^{(3)}(s_3) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} h_2(s_1) \\ h_2(s_2) \\ h_2(s_3) \end{bmatrix}$$

All hash functions are
$[N] \to \mathrm{GF}(2^p)$

M. Dietzfelbinger and R. Pagh, "Succinct data structures for retrieval and approximate membership," in *Int. Colloq. Automata, Languages, and Program.* Springer, 2008, pp. 385–396.
E. Porat, "An optimal bloom filter replacement based on matrix solving," in Int. Comput. Sci. Symp. Russia. Springer, 2009, pp. 263–273.

Linear Equations

$$\begin{bmatrix} h_1^{(1)}(s_1) & h_1^{(2)}(s_1) & h_1^{(3)}(s_1) \\ h_1^{(1)}(s_2) & h_1^{(2)}(s_2) & h_1^{(3)}(s_2) \\ h_1^{(1)}(s_3) & h_1^{(2)}(s_3) & h_1^{(3)}(s_3) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} h_2(s_1) \\ h_2(s_2) \\ h_2(s_3) \end{bmatrix}$$

All hash functions are $[N] \to \mathrm{GF}(2^p)$

Decoding:

$$h_1^{(1)}(s_k)z_1 + h_1^{(2)}(s_k)z_2 + h_1^{(3)}(s_k)z_3 = h_2(s_k) \Rightarrow \mathrm{ACK}$$

M. Dietzfelbinger and R. Pagh, “Succinct data structures for retrieval and approximate membership,” in *Int. Colloq. Automata, Languages, and Program.* Springer, 2008, pp. 385–396.
E. Porat, “An optimal bloom filter replacement based on matrix solving,” in Int. Comput. Sci. Symp. Russia. Springer, 2009, pp. 263–273.

Linear Equations

$$\begin{bmatrix} h_1^{(1)}(s_1) & h_1^{(2)}(s_1) & h_1^{(3)}(s_1) \\ h_1^{(1)}(s_2) & h_1^{(2)}(s_2) & h_1^{(3)}(s_2) \\ h_1^{(1)}(s_3) & h_1^{(2)}(s_3) & h_1^{(3)}(s_3) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} h_2(s_1) \\ h_2(s_2) \\ h_2(s_3) \end{bmatrix}$$

All hash functions are
$[N] \to \mathrm{GF}(2^p)$

Decoding:

$$h_1^{(1)}(s_k)z_1 + h_1^{(2)}(s_k)z_2 + h_1^{(3)}(s_k)z_3 = h_2(s_k) \Rightarrow \mathrm{ACK}$$

All we need to send is $[z_1 \quad z_2 \quad z_3]^T$
(assuming the solution exists)

Kp bits

M. Dietzfelbinger and R. Pagh, “Succinct data structures for retrieval and approximate membership,” in *Int. Colloq. Automata, Languages, and Program.* Springer, 2008, pp. 385–396.
E. Porat, “An optimal bloom filter replacement based on matrix solving,” in Int. Comput. Sci. Symp. Russia. Springer, 2009, pp. 263–273.

Linear Equations

$$\begin{bmatrix} h_1^{(1)}(s_1) & h_1^{(2)}(s_1) & h_1^{(3)}(s_1) \\ h_1^{(1)}(s_2) & h_1^{(2)}(s_2) & h_1^{(3)}(s_2) \\ h_1^{(1)}(s_3) & h_1^{(2)}(s_3) & h_1^{(3)}(s_3) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} h_2(s_1) \\ h_2(s_2) \\ h_2(s_3) \end{bmatrix}$$

All hash functions are
$[N] \to \mathrm{GF}(2^p)$

Decoding:

$$h_1^{(1)}(s_k)z_1 + h_1^{(2)}(s_k)z_2 + h_1^{(3)}(s_k)z_3 = h_2(s_k) \Rightarrow \mathrm{ACK}$$

All we need to send is $[z_1 \quad z_2 \quad z_3]^T$ (assuming the solution exists)

Kp bits

$$\varepsilon_{\mathrm{fp}} = 2^{-p} \Leftrightarrow p = \left\lceil \log_2\left(\frac{1}{\varepsilon_{\mathrm{fp}}}\right) \right\rceil$$

M. Dietzfelbinger and R. Pagh, “Succinct data structures for retrieval and approximate membership,” in *Int. Colloq. Automata, Languages, and Program.* Springer, 2008, pp. 385–396.

E. Porat, “An optimal bloom filter replacement based on matrix solving,” in Int. Comput. Sci. Symp. Russia. Springer, 2009, pp. 263–273.

Linear Equations

$$\begin{bmatrix} h_1^{(1)}(s_1) & h_1^{(2)}(s_1) & h_1^{(3)}(s_1) \\ h_1^{(1)}(s_2) & h_1^{(2)}(s_2) & h_1^{(3)}(s_2) \\ h_1^{(1)}(s_3) & h_1^{(2)}(s_3) & h_1^{(3)}(s_3) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} h_2(s_1) \\ h_2(s_2) \\ h_2(s_3) \end{bmatrix}$$

All hash functions are $[N] \to \mathrm{GF}(2^p)$

Decoding:

$$h_1^{(1)}(s_k)z_1 + h_1^{(2)}(s_k)z_2 + h_1^{(3)}(s_k)z_3 = h_2(s_k) \Rightarrow \mathrm{ACK}$$

All we need to send is $[z_1 \quad z_2 \quad z_3]^T$ (assuming the solution exists)

Kp bits

$$\varepsilon_{\mathrm{fp}} = 2^{-p} \Leftrightarrow p = \left\lceil \log_2\left(\frac{1}{\varepsilon_{\mathrm{fp}}}\right) \right\rceil$$

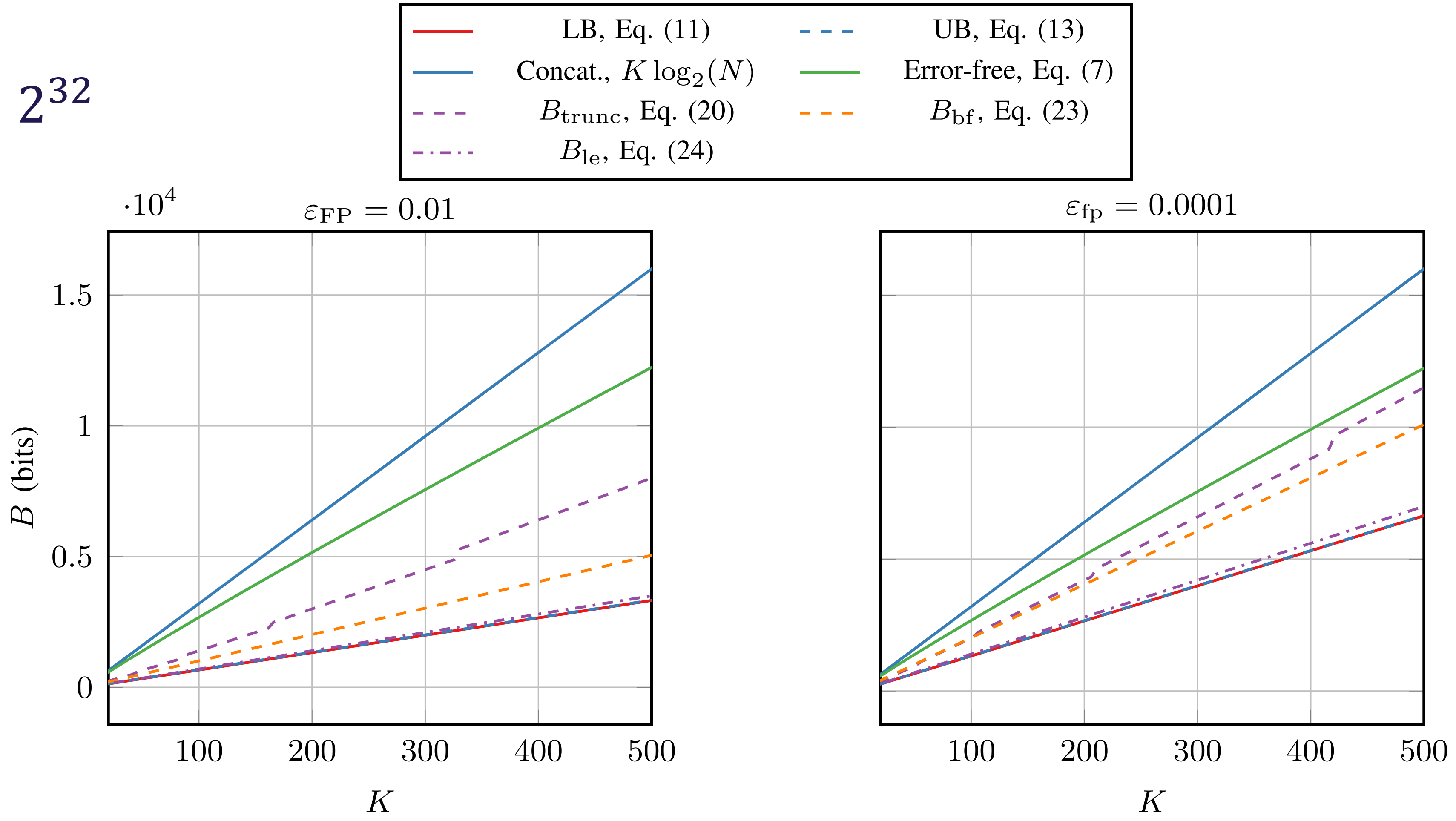
Recall the bound:

$$B_{\mathrm{fp}}^* = K \log_2\left(\frac{1}{\varepsilon_{\mathrm{fp}}}\right) \pm \mathcal{O}(\log\log N)$$

M. Dietzfelbinger and R. Pagh, "Succinct data structures for retrieval and approximate membership," in *Int. Colloq. Automata, Languages, and Program.* Springer, 2008, pp. 385–396.
E. Porat, "An optimal bloom filter replacement based on matrix solving," in Int. Comput. Sci. Symp. Russia. Springer, 2009, pp. 263–273.

Comparison

$N = 2^{32}$



Outline

Part 1
Information Theoretic Bounds

Part 2
Practical Schemes

Part 3
Applications in ARQ protocols

Downlink Erasure Channel

ACK encoder

$$f: \binom{[N]}{K} \to \{0,1\}^B$$

User n's channel

$$X \in \{0,1\}^B \to X, \quad \mathrm{e}$$

User n's decoder

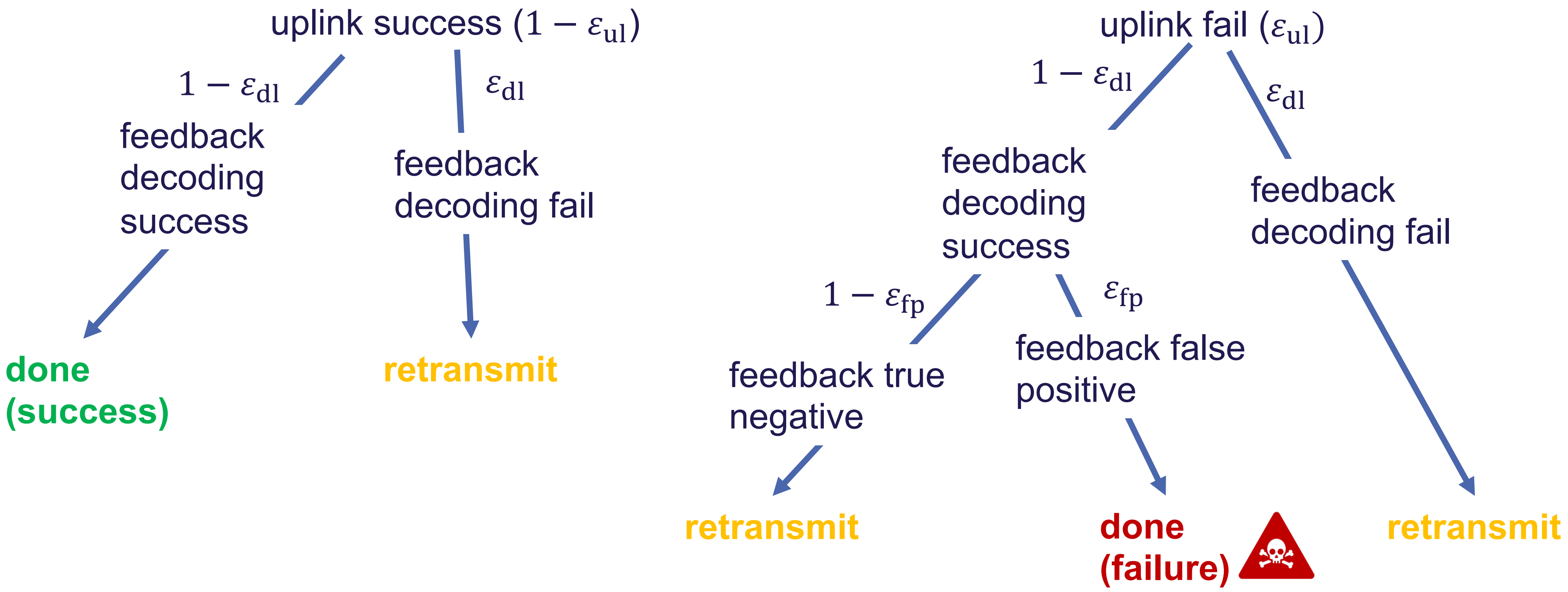
$$g_n: \{0,1\}^B \cup \mathrm{e} \to \{\mathrm{ACK}, \mathrm{NACK}\}$$

Erasure probability assumed to be equal to the outage probability

For evaluation we will assume:

- Poisson arrivals
- Fixed-length coding
- Rayleigh fading
- 2048 symbols
- 64 tx antennas (but no precoding)

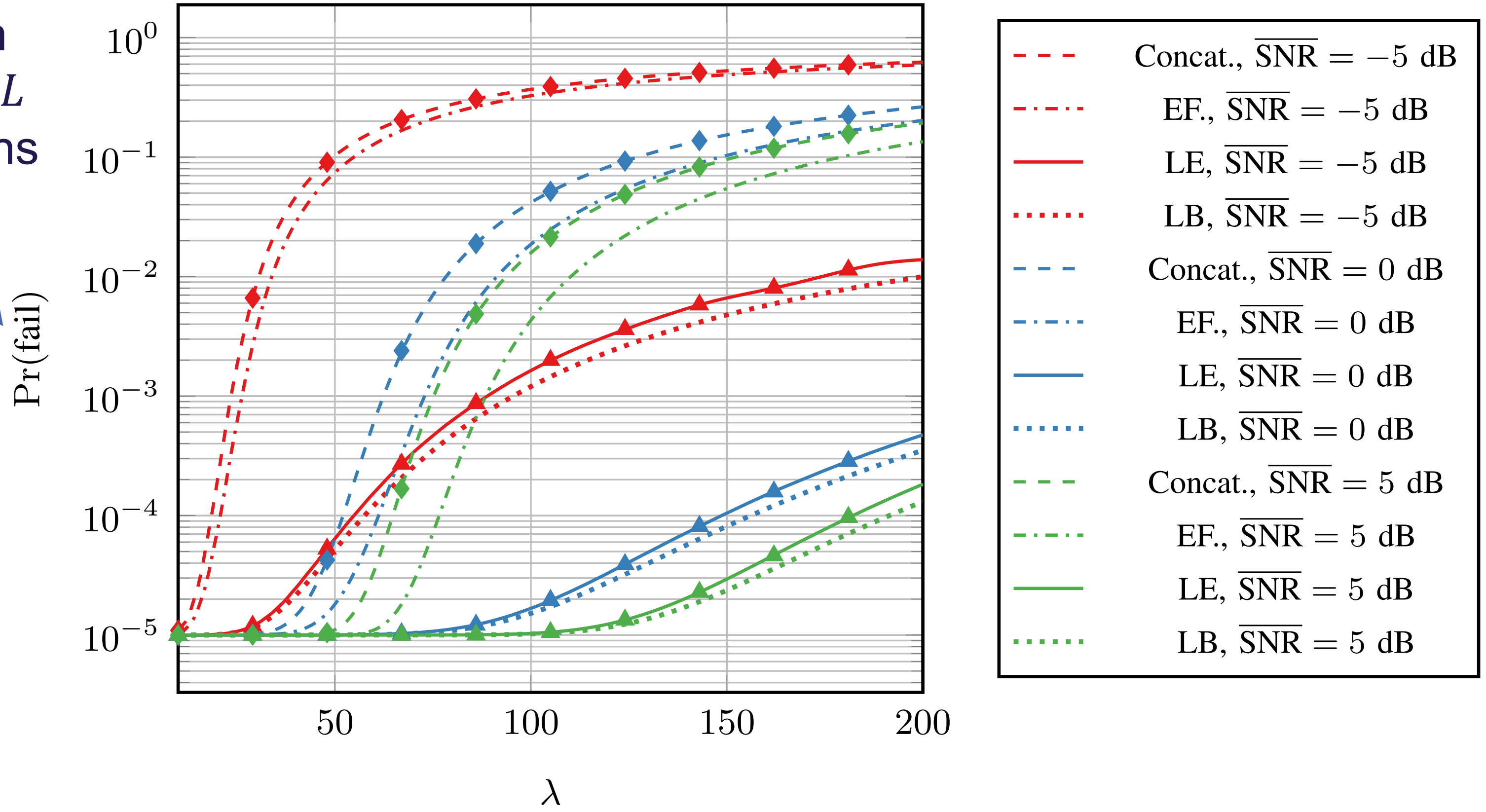
ARQ Model



Reliable feedback is a trade-off between reliable transmission and false positive probability

Fixed-length Feedback with Fading

Not in green state within L transmissions



- $L = 5$
- $\varepsilon_{\text{ul}} = 0.1$
- $K \sim \text{Poisson}(\lambda)$ (iid in each retransmission)
- Rayleigh fading
- 2048 symbols
- 64 tx antennas
- Markers indicate simulations

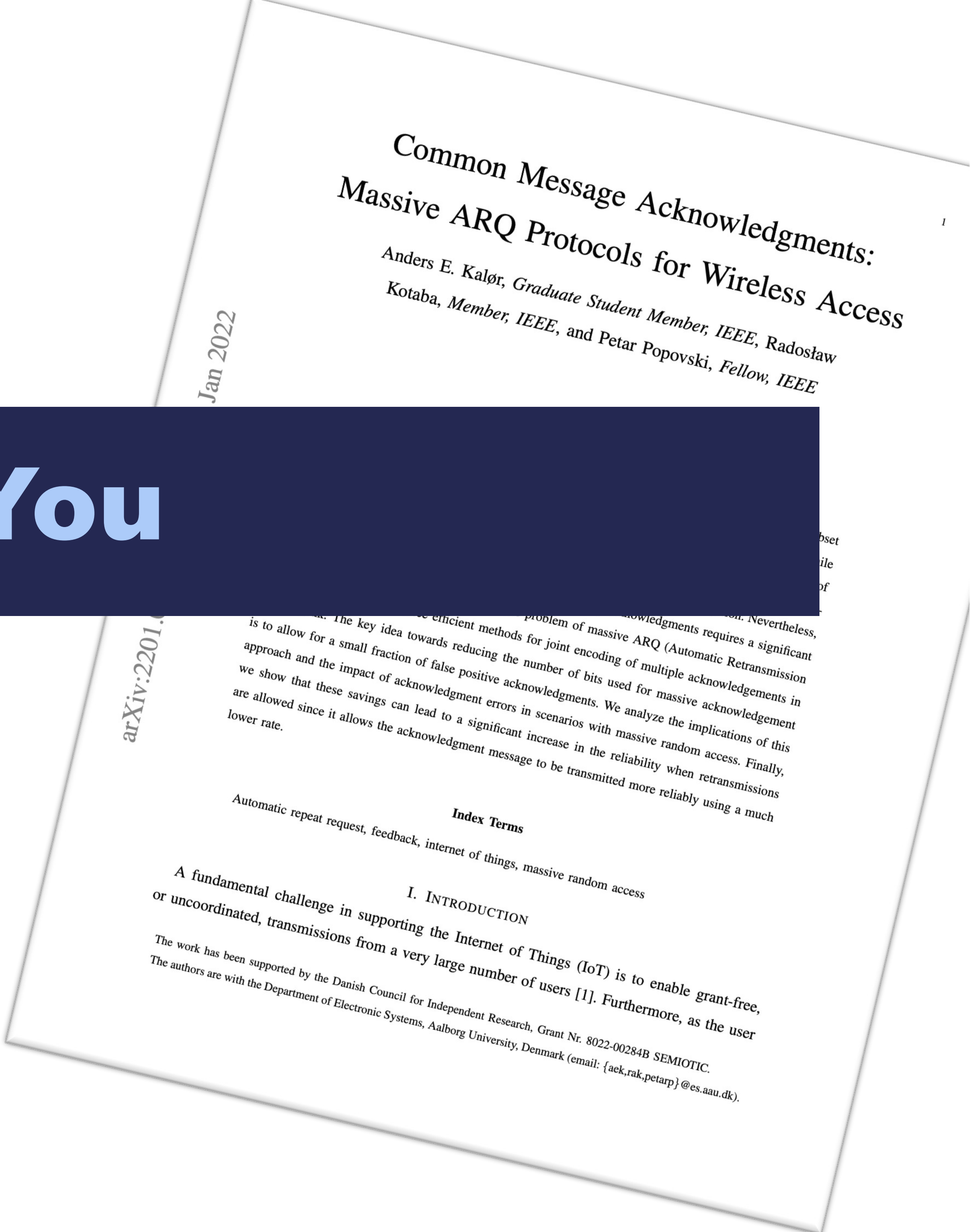
- More efficient coding allows for lower transmission rate
- Significantly higher reliability despite false positives

Resolving Failures

- Some users **erronously believe they succeeded** when they fail
- False positives exist in all ARQ systems (CRC failures, etc.)
  - Example: 16-bit CRC gives $\varepsilon_{\mathrm{fp}} \approx 1.5 \cdot 10^{-5}$
  - ACK messages are usually designed to have $\varepsilon_{\mathrm{fp}} \ll \varepsilon_{\mathrm{fn}}$, but we do the opposite
- Need to be resolved at higher layers, e.g., using sequence numbers

Conclusions

- Acknowledgment feedback in massive random access is nontrivial
- Identifier concatenation is highly sub-optimal
- Allowing for false positive errors significantly reduces the number of bits required
- This leads to significant ARQ reliability gains despite false positives



Thank You

A. E. Kalør, R. Kotaba and P. Popovski, "**Common Message Acknowledgments: Massive ARQ Protocols for Wireless Access**," in *IEEE Transactions on Communications*, vol. 70, no. 8, pp. 5258-5270, Aug. 2022.