STAIR Codes: A General Family of Erasure Codes for Tolerating Device and Sector Failures in Practical Storage Systems

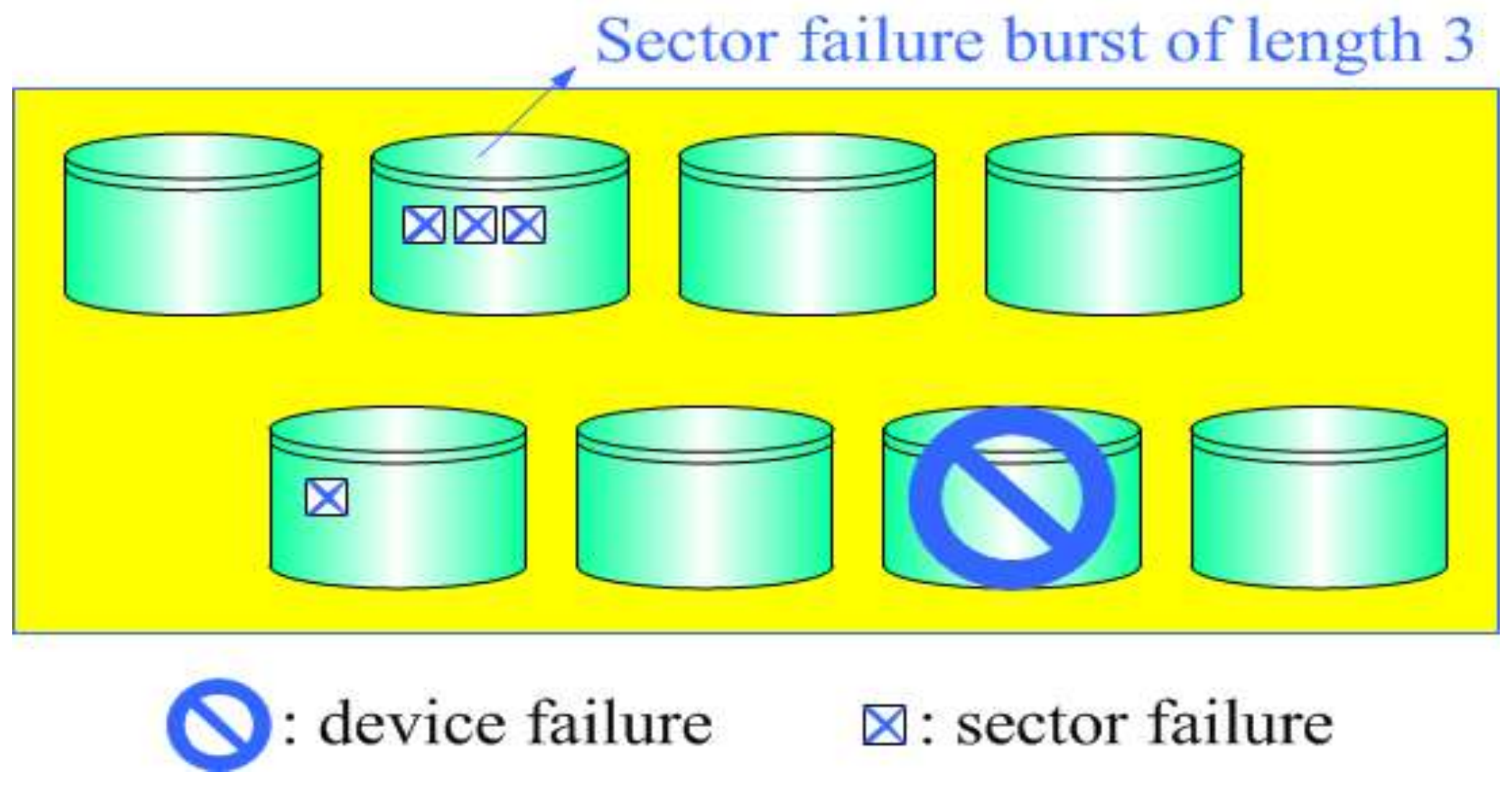
Mingqiang Li and Patrick P. C. Lee

The Chinese University of Hong Kong

Source code available at http://ansrlab.cse.cuhk.edu.hk/software/stair

Goals

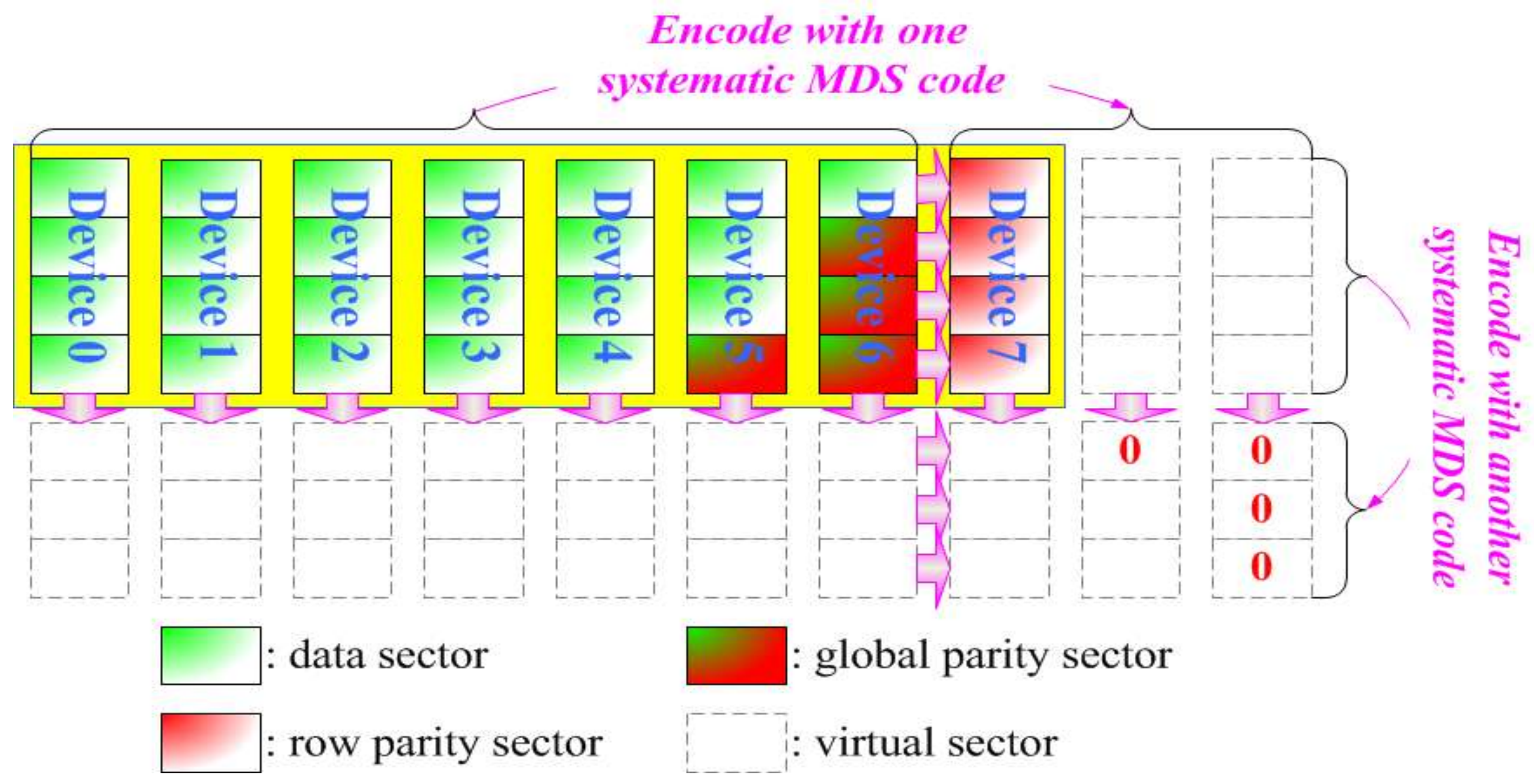
- Construct space-efficient erasure codes to tolerate simultaneous **device failures** and **sector failures**
- Provide a **general** construction *without restriction* on
  - size of a storage array,
  - number of tolerable device failures, or
  - number of tolerable sector failures.



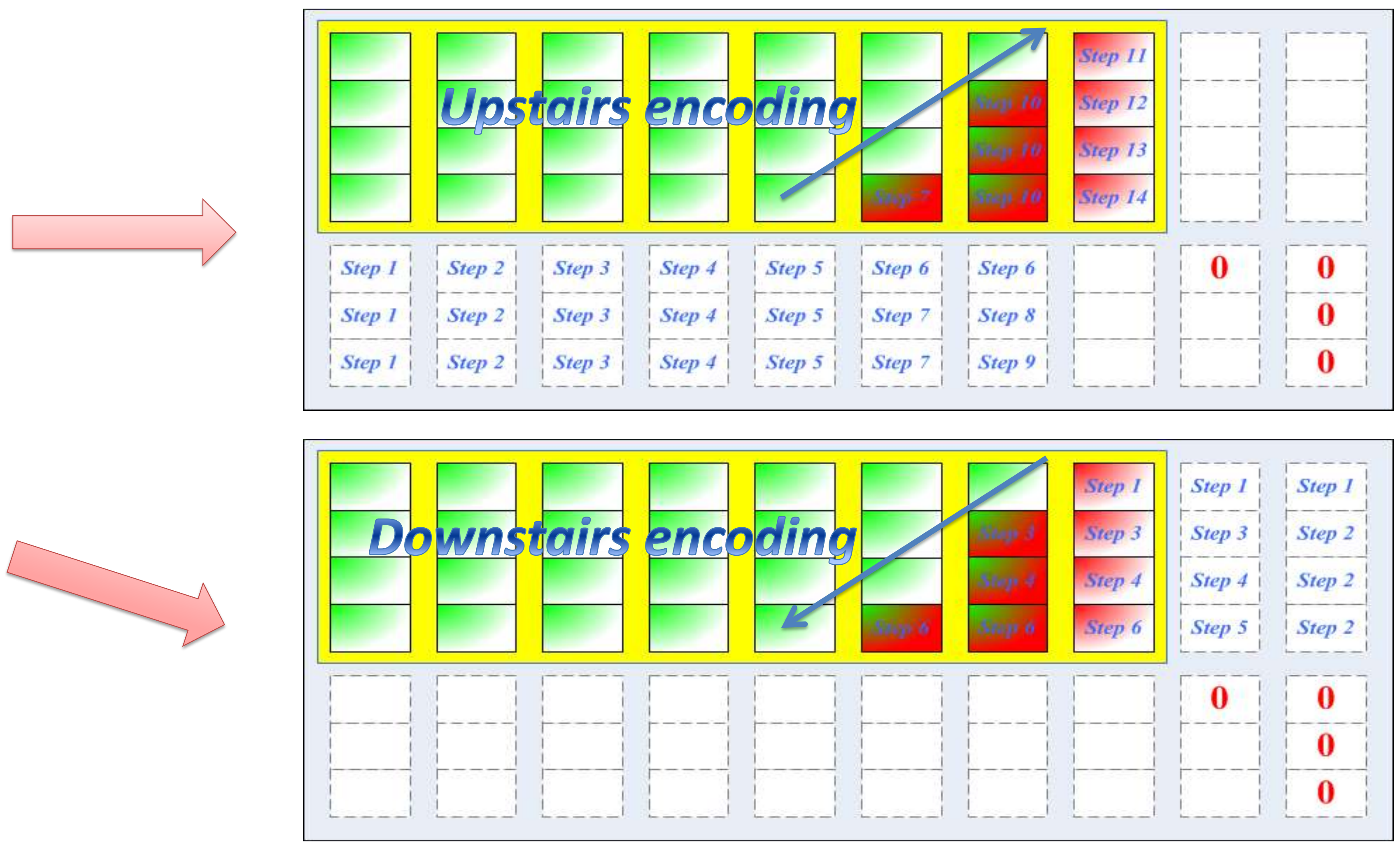
A mixed failure scenario

Design

- Define *sector failure coverage vector* **e**:
  - Bound number of devices with sector failures
  - Bound number of sector failures per device
- Run **two encoding phases** with *systematic MDS codes* (e.g. RS codes);
- Propose **upstairs encoding** and **downstairs encoding** methods, which *provide complementary performance advantages for different configurations*

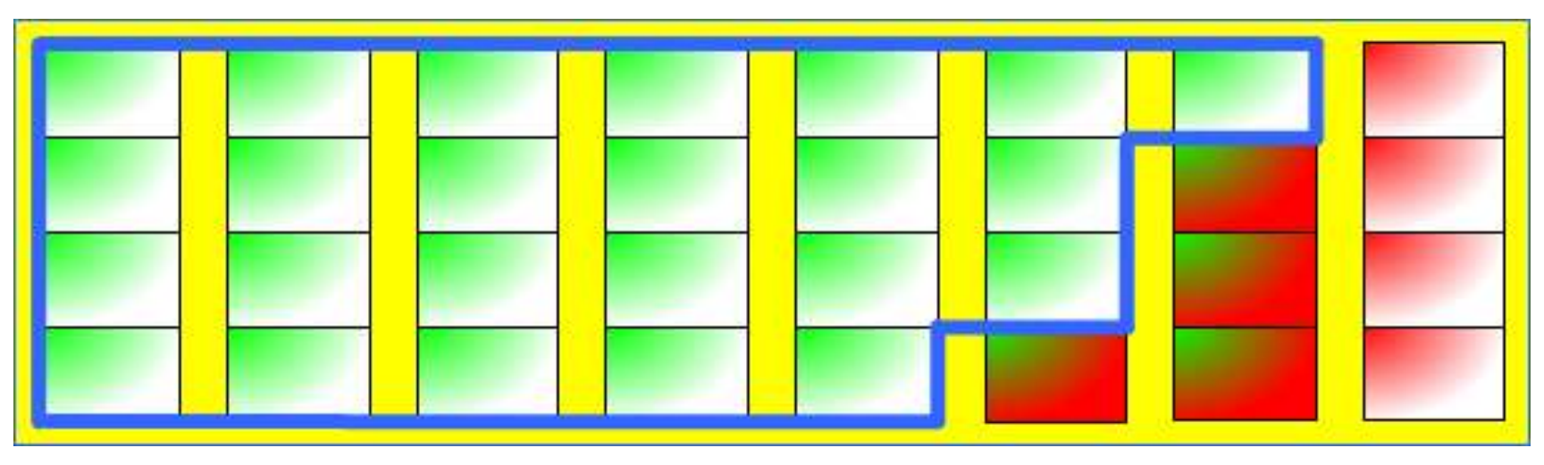


A STAIR code with **row parity sectors** to tolerate *m*=1 device failure and **global parity sectors** to tolerate **e**=(1,3) sector failures

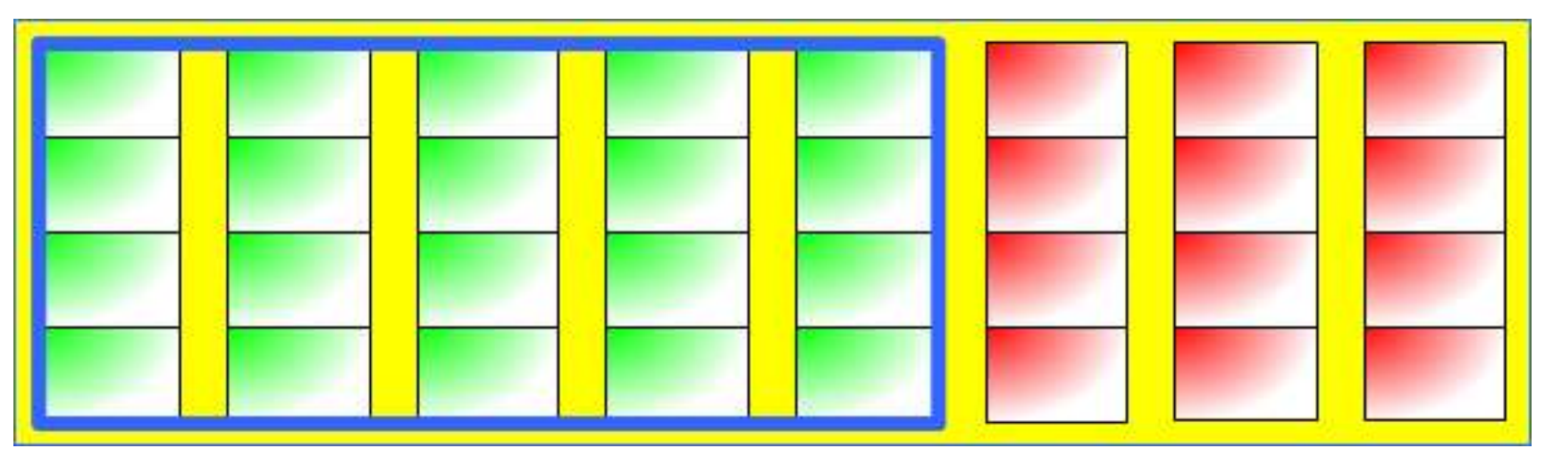


STAIR Codes vs. Existing Coding Schemes

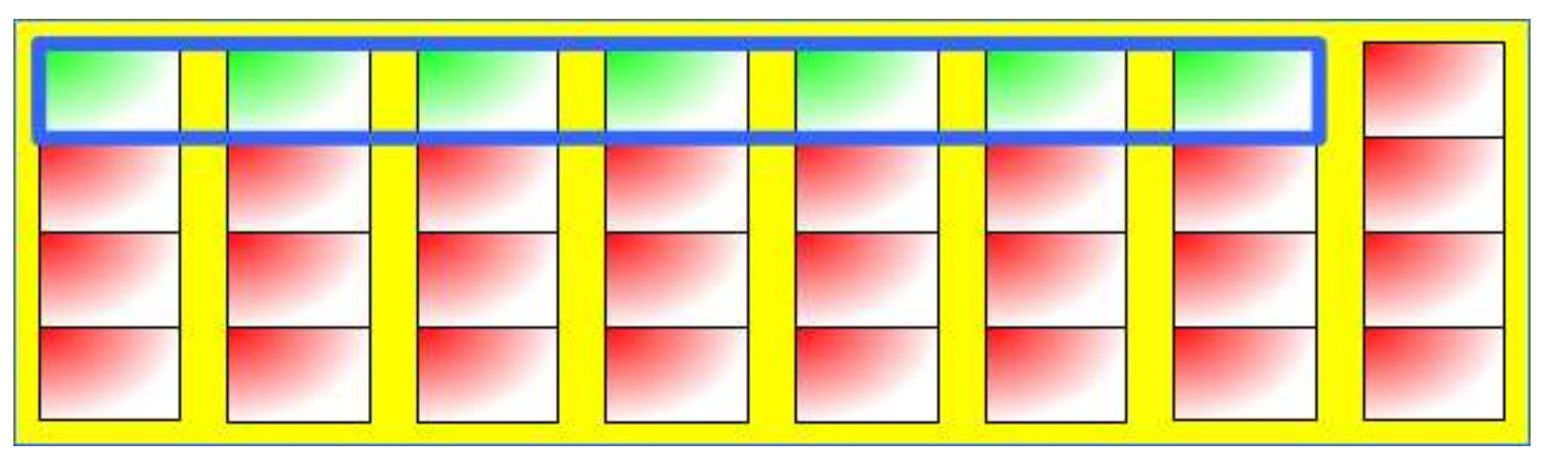
- STAIR codes improve space efficiency significantly over RS codes and IDR schemes:



STAIR: 24/32



RS: 20/32



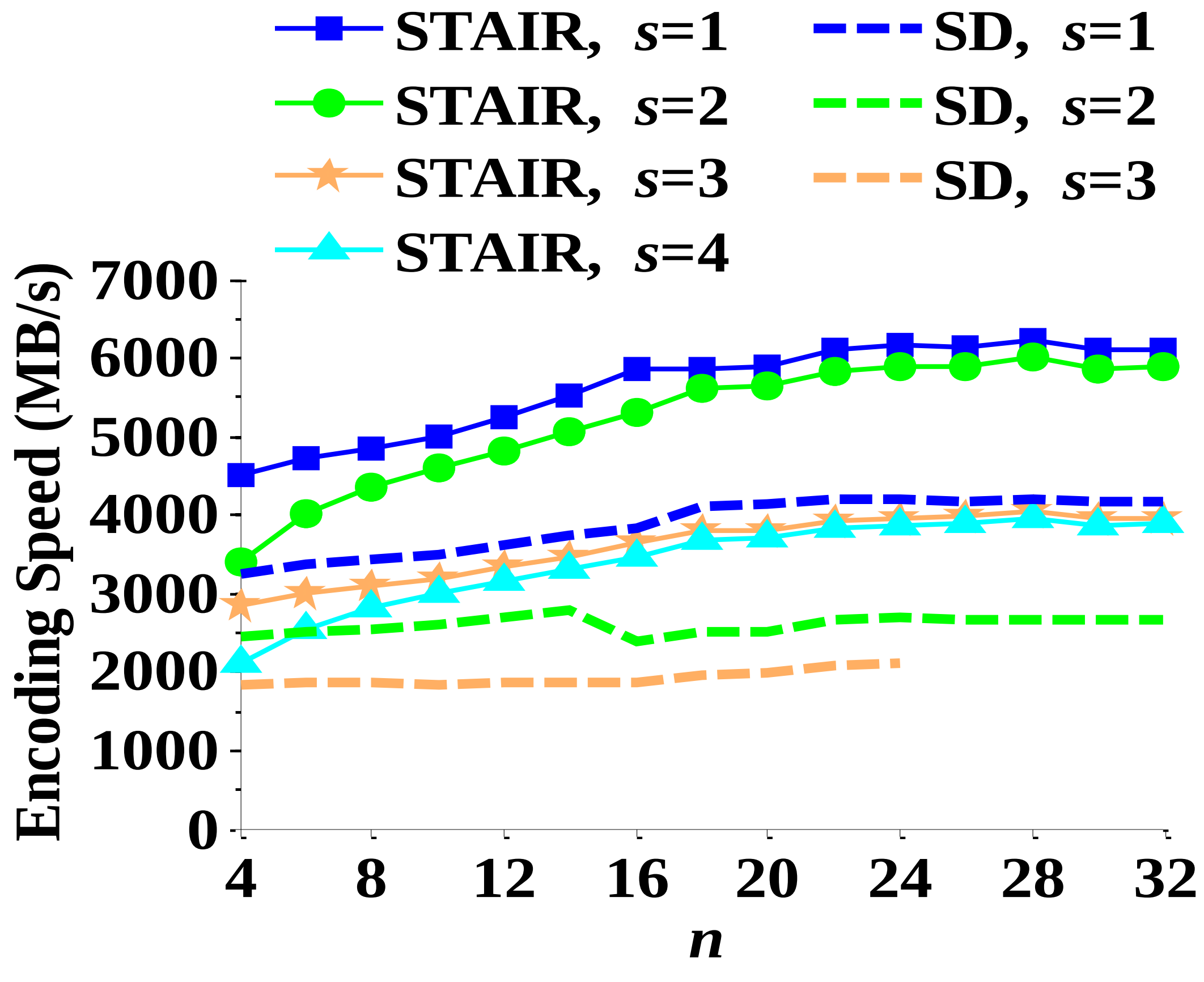
IDR: 7/32

- STAIR codes support more general construction than SD codes [Plank, FAST'13] (limited to ≤ *3 sector failures*)

Practical Properties of STAIR Codes

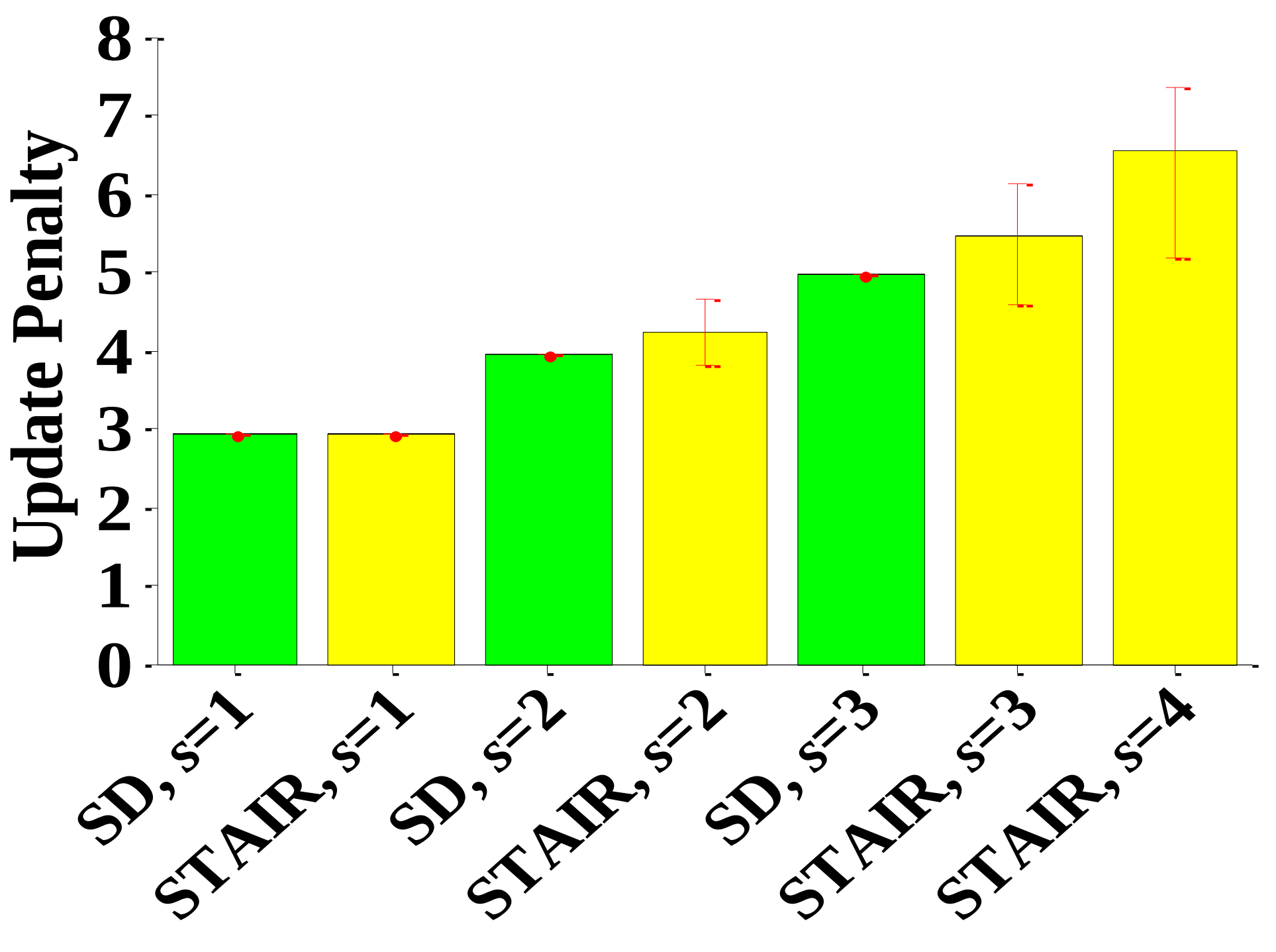
1. Compared to SD codes, STAIR codes **improve the encoding speed**, due to *parity reuse* in upstairs/downstairs encoding;
2. When only device failures occur, both STAIR and SD codes have **identical decoding performance to RS codes**;
3. Both STAIR and SD codes **have high update cost**, due to the additional updates of global parity sectors

Test environment: Intel Core i5-3570 CPU at 3.40GHz with SSE4.2 support



When *r*=16 and *m*=1

n: # of devices in a storage array;
r: # of sectors per chunk;
m: # of tolerable device failures;
s: # of tolerable sector failures.



When *n*=16, *r*=16, and *m*=1