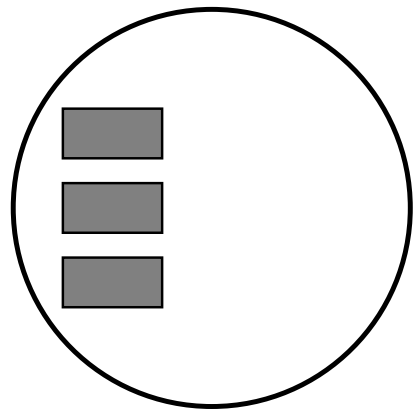
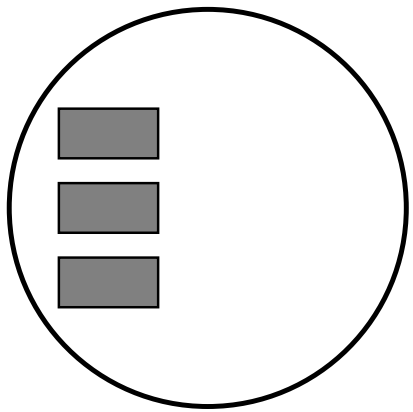
Distributed random network coding

- **Initialization:**
  - Source node stores block of message packets in its memory



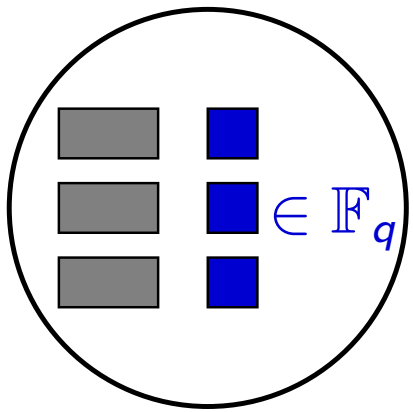
Distributed random network coding

- **Initialization:**
  - Source node stores block of message packets in its memory
- **Operation:**
  - When a node receives a packet
    - node stores packet in its memory
  - When a node injects a packet
    - node forms packet from random linear combination of packets in its memory



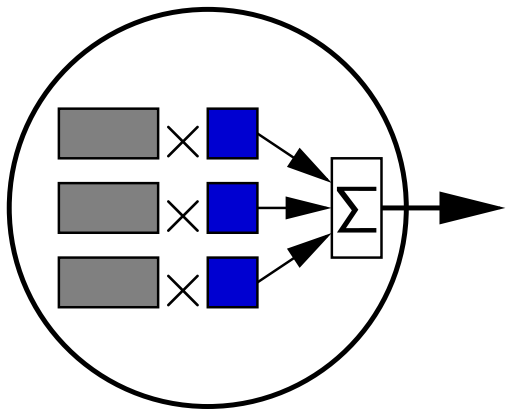
Distributed random network coding

- **Initialization:**
  - Source node stores block of message packets in its memory
- **Operation:**
  - When a node receives a packet
    - node stores packet in its memory
  - When a node injects a packet
    - node forms packet from random linear combination of packets in its memory



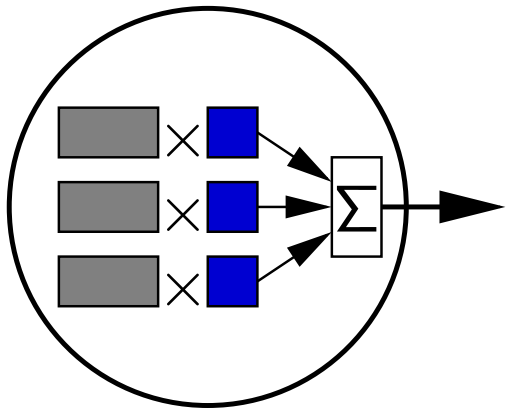
Distributed random network coding

- **Initialization:**
  - Source node stores block of message packets in its memory
- **Operation:**
  - When a node receives a packet
    - node stores packet in its memory
  - When a node injects a packet
    - node forms packet from random linear combination of packets in its memory



Distributed random network coding

- **Initialization:**
  - Source node stores block of message packets in its memory
- **Operation:**
  - When a node receives a packet
    - node stores packet in its memory
  - When a node injects a packet
    - node forms packet from random linear combination of packets in its memory
- **Decoding:**
  - Sink nodes perform Gaussian elimination



This scheme is capacity-achieving

- We require packet receptions to occur with an average rate
- Source of losses is immaterial
  - Buffer overflow
  - Link outage
  - Collision
- Reference: Lun et al. (submitted to *Trans. IT*)

What can be improved?

- Three things:
  1. Computational complexity
  2. Memory requirements
  3. Transmission overhead
- Reference: Pakzad et al. (*ISIT 2005*)