# Raptor codes
# Application Layer FEC

Michael Luby

January 30, 2012
ICNC

# Digital media delivery challenges today

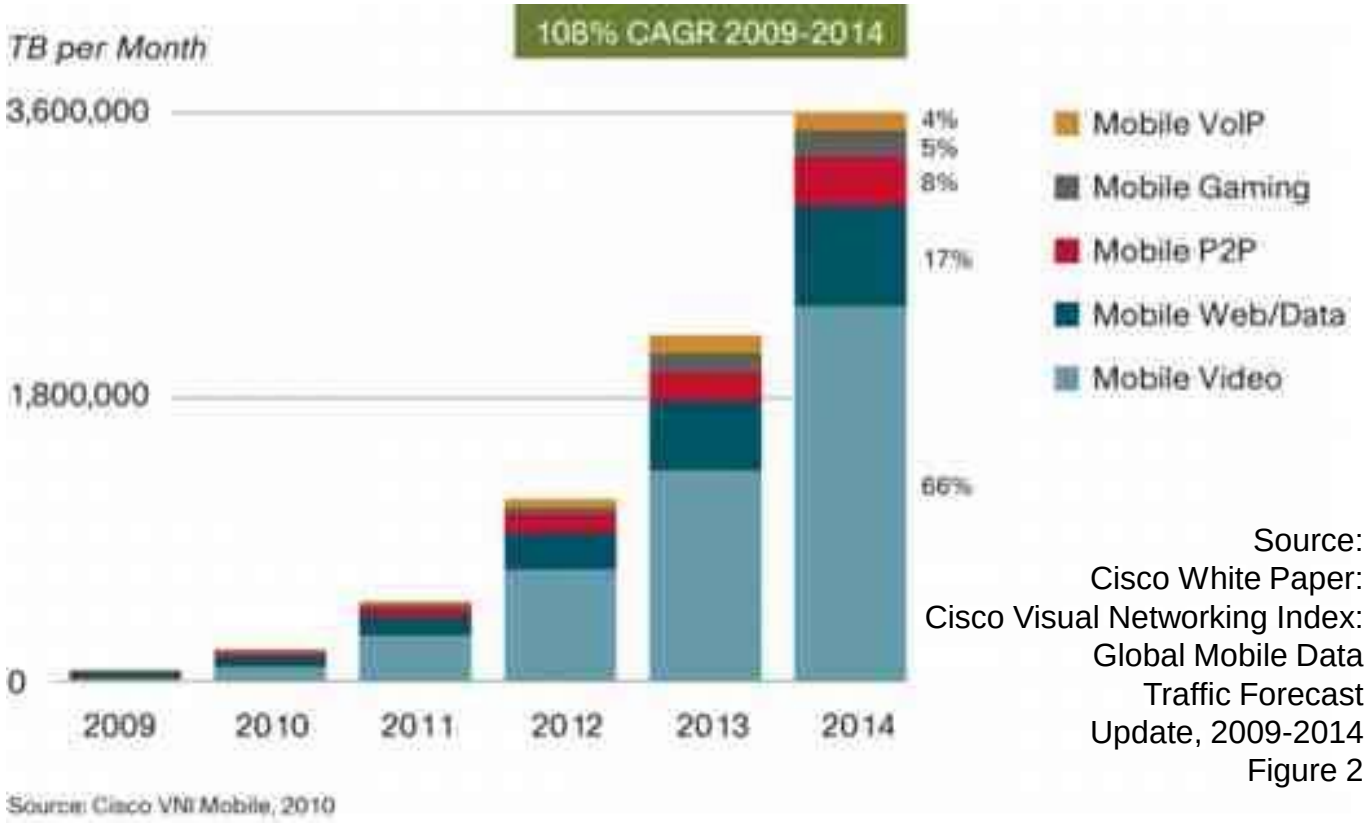
- Reliable and efficient content delivery is increasingly important
  - Video streaming
  - Multimedia content delivery
- High value need for efficient streaming and content delivery methods across all networks (wireless and wired) and device types (servers, laptops, handhelds, etc.)
- Content providers require consistent high quality-of-service (QOS) across all their distribution methods, delivered economically
  - Mobile and multimedia content delivery are booming
  - Content delivery over 3G, 4G, Wi-Fi increasing dramatically
  - Broadcast over eMBMS is a potential emerging market
- Some applications require complex synchronization between different networks to support multi-path data delivery

# The Mobile Video Challenge

- The mobile video landscape
  - Mobile Internet use is dramatically expanding
  - Video traffic is growing exponentially & is a large fraction of the usage
- The challenges
  - Mobile users expect high quality video **experience**
  - Network operators need to offer quality experience affordably

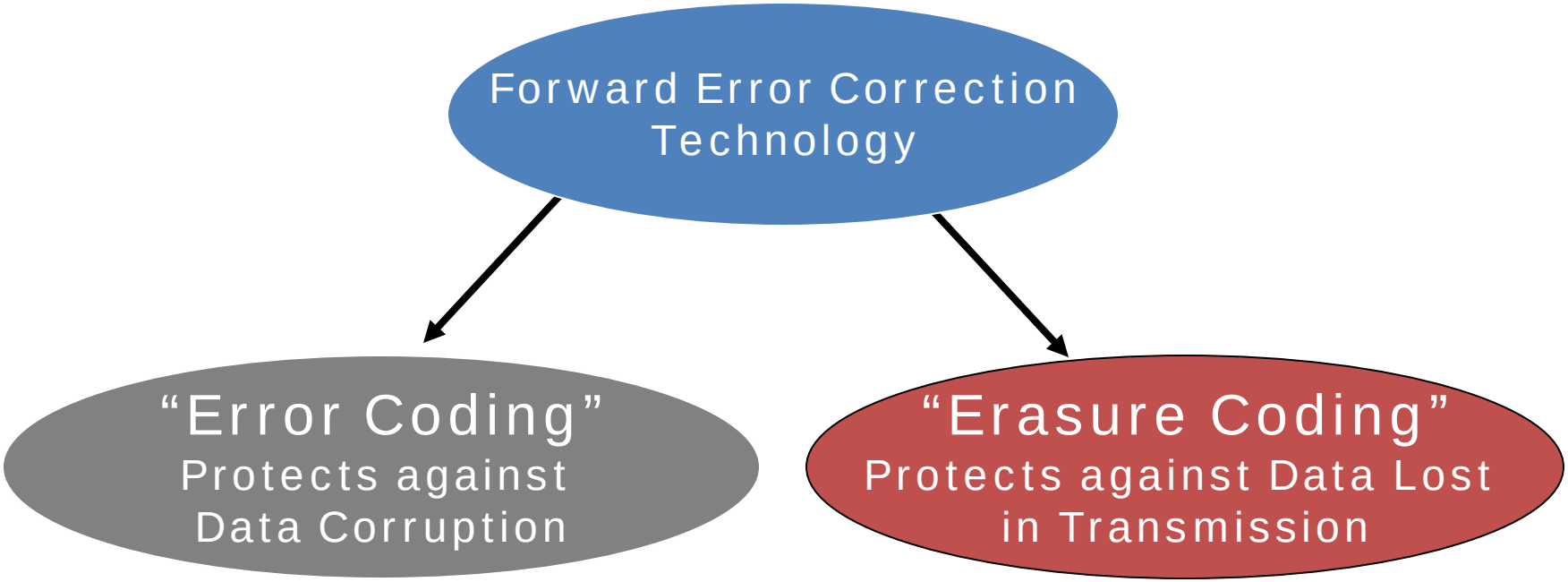
39 times growth of mobile data

66% mobile video by 2014




Source:
Cisco White Paper:
Cisco Visual Networking Index:
Global Mobile Data
Traffic Forecast
Update, 2009-2014
Figure 2

# Raptor technology complements traditional error coding



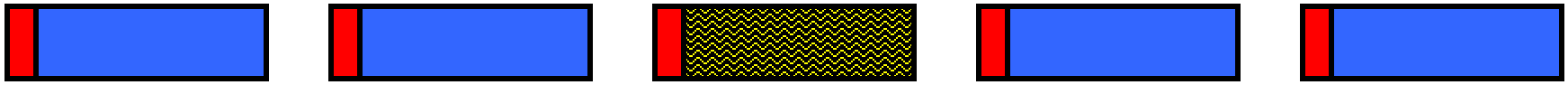

- **Vast majority of current use of FEC**
- **Probably what you're familiar with**
- **Typically applied at layers 1 or 2**
- **Usually performed in hardware**
- **PHY-FEC (physical layer FEC)**

- **Commercial application relatively new**
- **Applied above layer 2**
- **Complement to Error Coding**
- **Typically performed in software**
- **AL-FEC (application layer FEC)**

# Packet transmission

**Packet payload**

**Packet header**

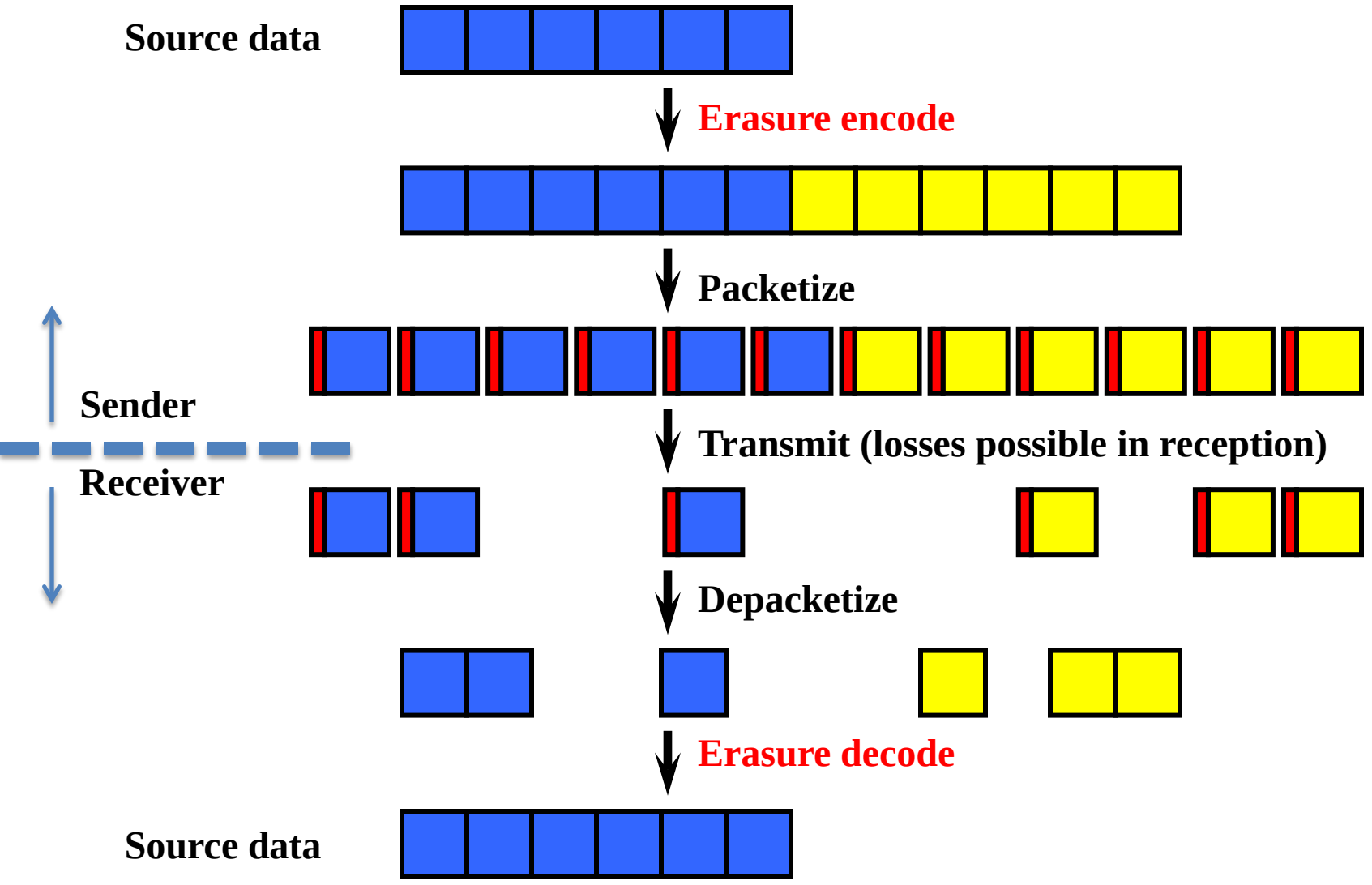


**Stream of packets**

**Received corrupted packet is discarded**

**Can identify received packet payloads from packet headers**

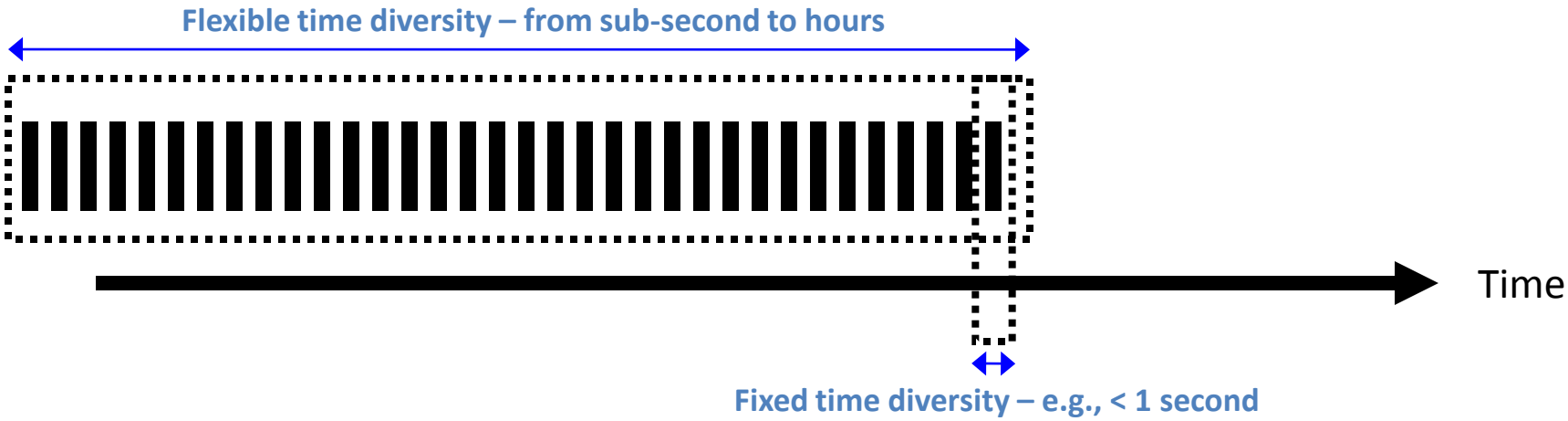
# Erasure Codes and AL-FEC



# AL-FEC and PHY-FEC are complementary

## AL-FEC

Packet loss protection over small to large block
Flexible time diversity
Flexible amount of protection

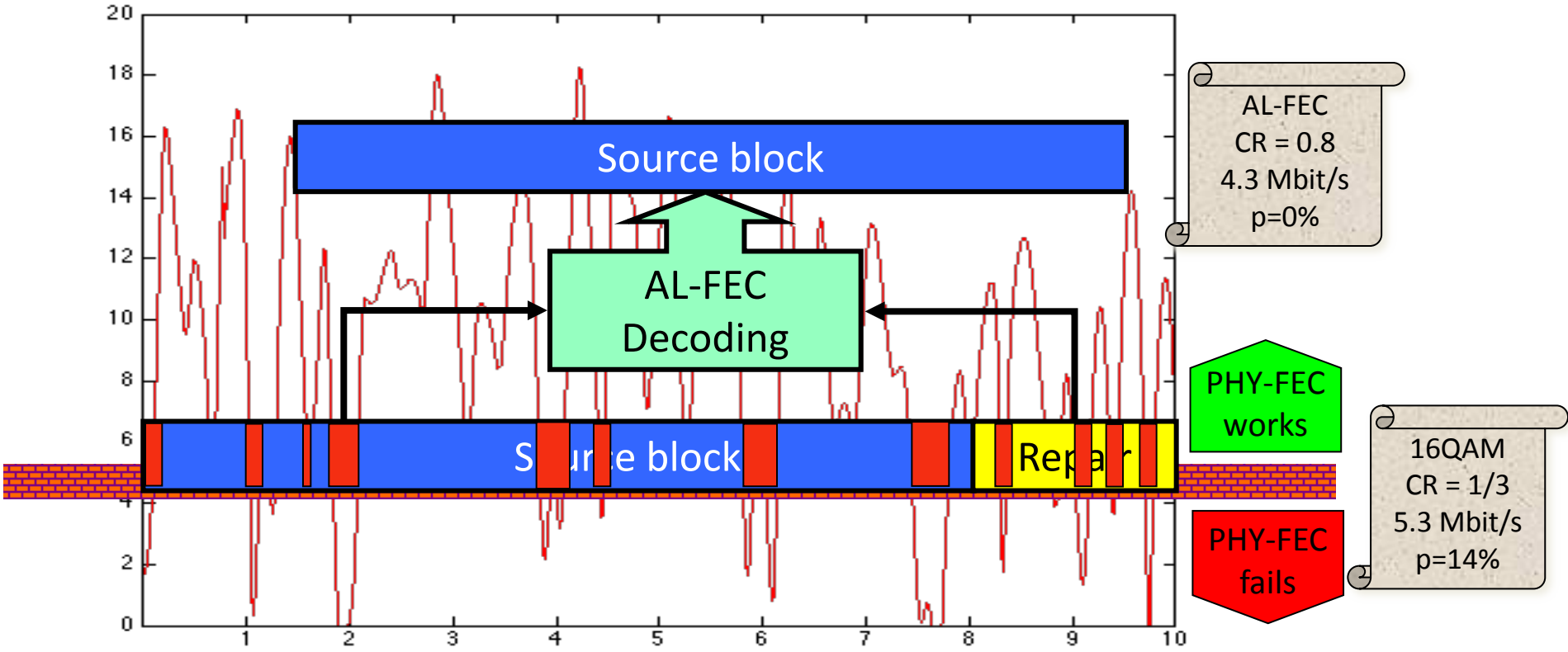



## PHY-FEC

Correct or discard corrupted packet data over small block
Fixed time diversity
Fixed amount of protection

# AL-FEC and PHY-FEC working together

- PHY-FEC corrects noise and interference
- AL-FEC "interleaves" and corrects erasures
  - Longer block length ("interleavers") ⎩ better performance



# Erasure Coding Approaches

- Reed-Solomon codes (1960, Reed and Solomon)
  - Based on **finite fields**
  - Systematic (strongly systematic)
  - **Has zero reception overhead**
  - Quadratic time in block size encoding and decoding (in practice)

# Erasure Coding Approaches

- LDPC codes (1960, Gallager)
  - Based on **regular random graphs**
  - Systematic (but not strongly systematic)
  - Incurs a significant relative reception overhead for all block sizes
  - **Linear time in block size encoding and decoding**

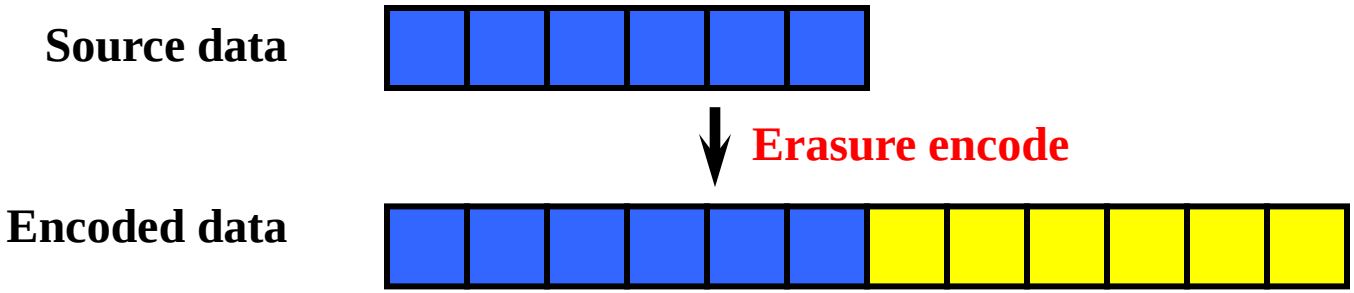
# Erasure Coding Approaches

- Tornado codes (1997, Luby et. al.)
  - Based on **irregular random graphs** (the first irregular LDPC codes)
  - Systematic (but not strongly systematic)
  - **Approaches zero relative reception overhead as block size grows**
  - **Linear time in block size encoding and decoding**
  - 2002 IEEE Information Theory Society paper award … for design and analysis of irregular LDPC error-correcting codes

# Erasure Coding Approaches

- LT codes (1998, Luby)
  - The first practical **fountain code**
  - Simpler irregular graph structure than Tornado
  - Non-systematic
  - Approaches zero relative reception overhead with growing block size
  - Almost linear time in block size encoding and decoding
  - 2009 ACM SIGCOMM Test of Time Award – fountain code approach to reliable distribution of bulk data … outstanding paper whose contents are still a vibrant and useful contribution today

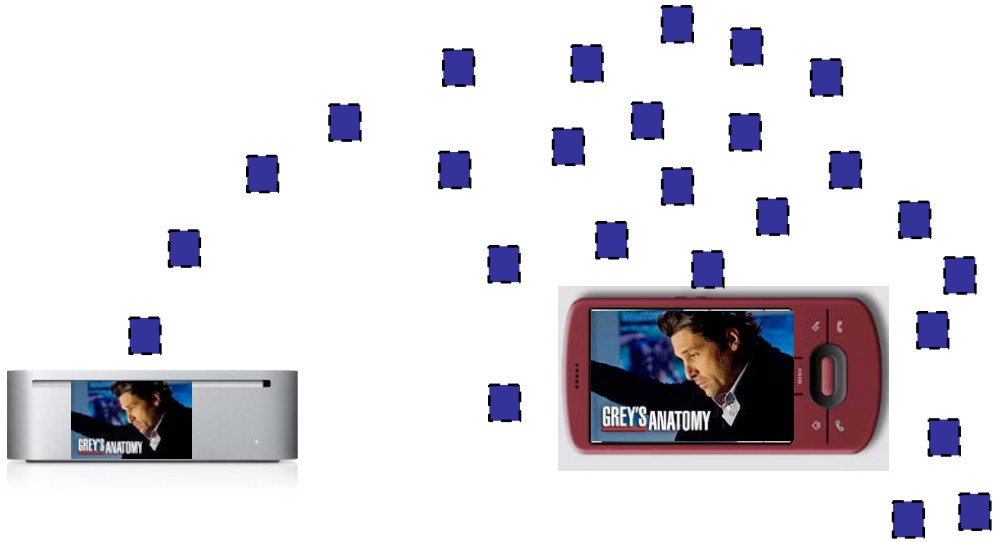
# Erasure codes with a rate



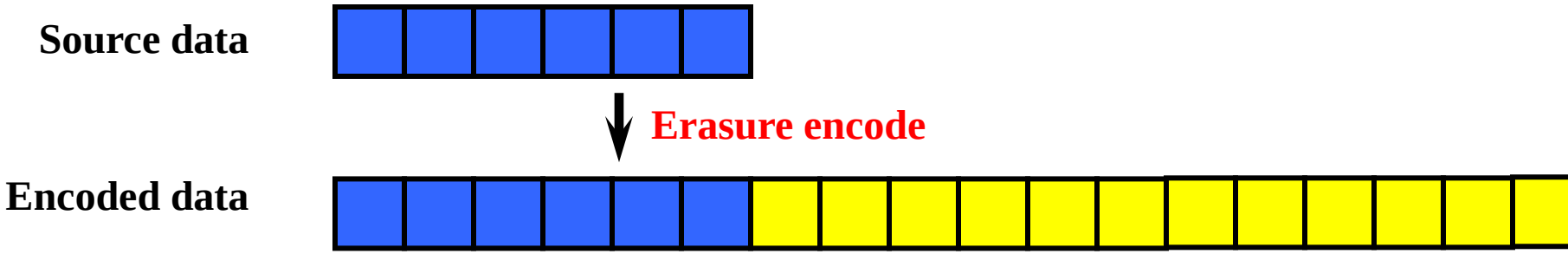

- Code rate = size of source data / size of encoded data
- For non-fountain codes, for a fixed source data size
  - Erasure code design is specifically for a given code rate
  - Each code rate uses a different erasure code design
  - Particular encoded symbols are generated differently for different code rate
  - Number of encoded symbols that can be generated is fixed by the design

# What is a fountain code?

- Generate as much encoding as desired
- Recover source from the minimal possible encoding
  - It doesn't matter what is received or lost
  - It only matters that enough is received



# Erasure codes without a rate – fountain codes




- Fountain codes have no predetermined rate
- For fountain codes, for a fixed source data size
  - Erasure code design is extendable to provide any code rate
  - All code rates use the same extendable erasure code design
  - Particular encoded symbols are generated independently of one another
  - Number of encoded symbols that can be generated on the fly is unconstrained

# Erasure Coding Approaches

- Raptor codes (2001, Shokrollahi, et. al.)
  - Based on LT codes
  - Systematic (strongly systematic)
  - Fountain codes
  - Approaches zero relative reception overhead as block size grows
  - **Excellent recovery properties** for all block sizes
  - **Linear time** in block size encoding and decoding
  - 2007 IEEE Comm. Society/Information Theory Society paper award … Raptor codes

# Raptor codes in standards

- Raptor codes (IETF RFC 5053)
  - Systematic fountain codes
  - Linear time encoding and decoding
  - Standardized – 3GPP MBMS, DVB-H IPDC
  - Good recovery properties – like a random code over GF(2)
  - Good flexibility
    - Up to 8,192 source symbols
    - Up to 65,384 source + repair symbols

- RaptorQ codes (IETF RFC 6330)
  - Systematic fountain codes
  - Linear time encoding and decoding
  - Proposal to all future standards (IEEE P2200, 3GPP eMBMS, …)
  - Great recovery properties – like a random code over GF(256)
  - Great flexibility
    - Up to 56,403 source symbols
    - Up to 16,777,216 source + repair symbols (essentially unlimited)

# How Raptor codes work

- Raptor codes are fountain codes
  - Can efficiently generate as much encoding as desired, on the fly
  - Doesn’t matter what is received or lost, only matters that enough is received
  - Can dynamically change the number of repair packets based on loss level
- Raptor codes are systematic codes
  - Source code is part of the encoding
- Raptor codes are used in an application specific manner
  - For content delivery – encode the entire file as a block or a set of sub-blocks if receiver memory is limited
  - For streaming – encode blocks of the stream
  - Also can be used at MAC layer to protect all data

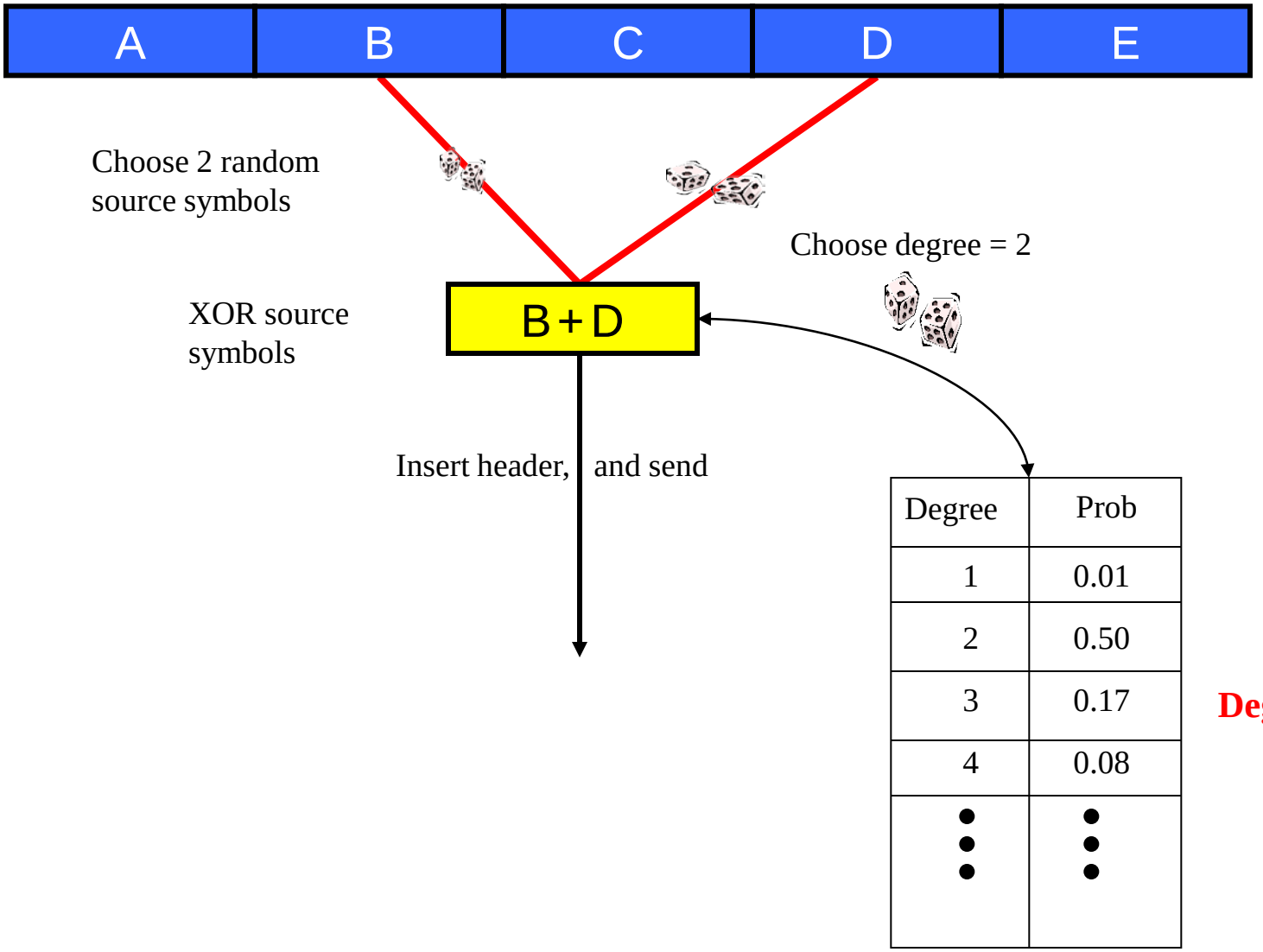
# LT encoding

Source block

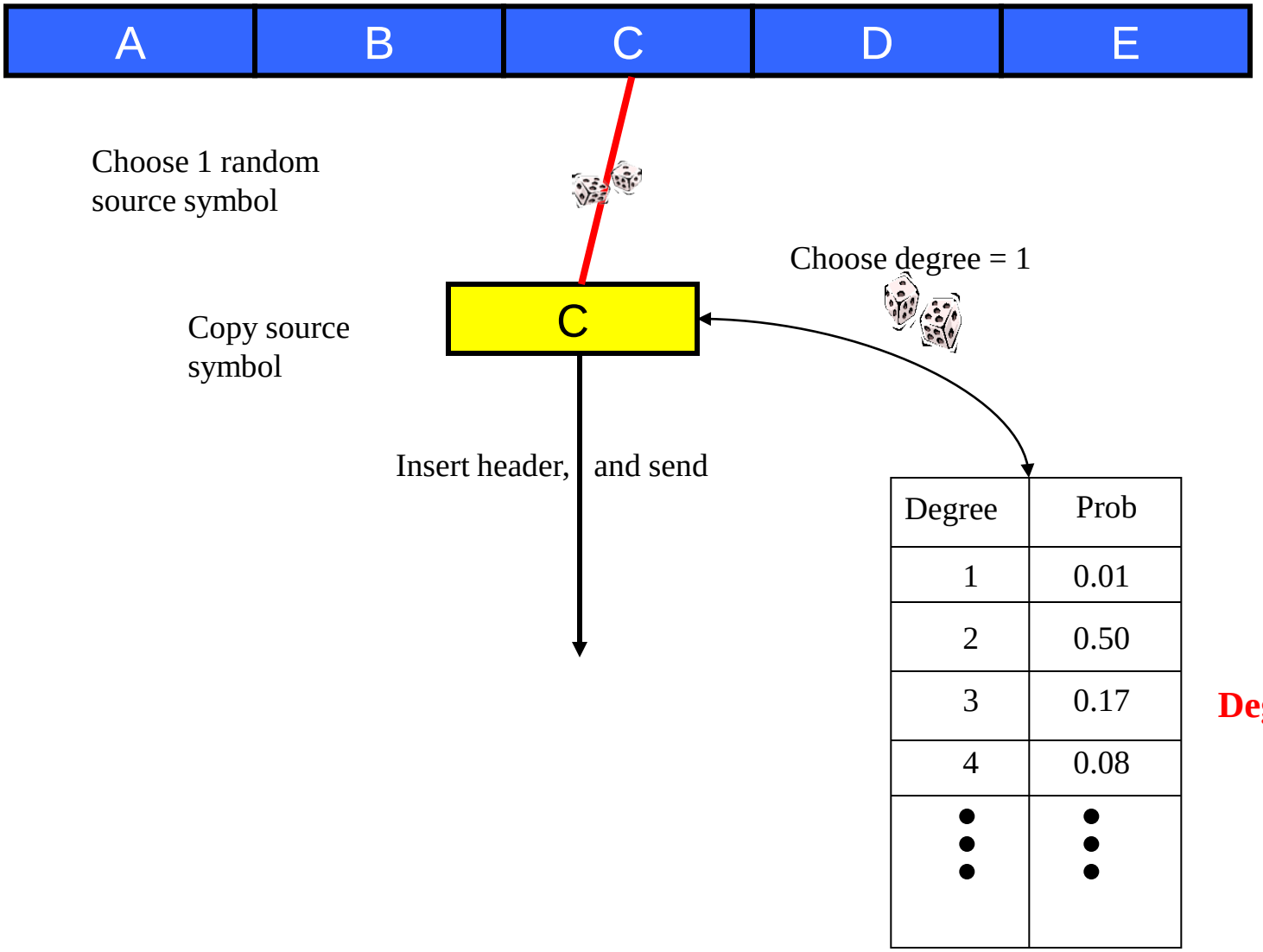
| Degree | Prob |
|---|---|
| 1 | 0.01 |
| 2 | 0.50 |
| 3 | 0.17 |
| 4 | 0.08 |
| ⋮ | ⋮ |

**Degree Distribution**

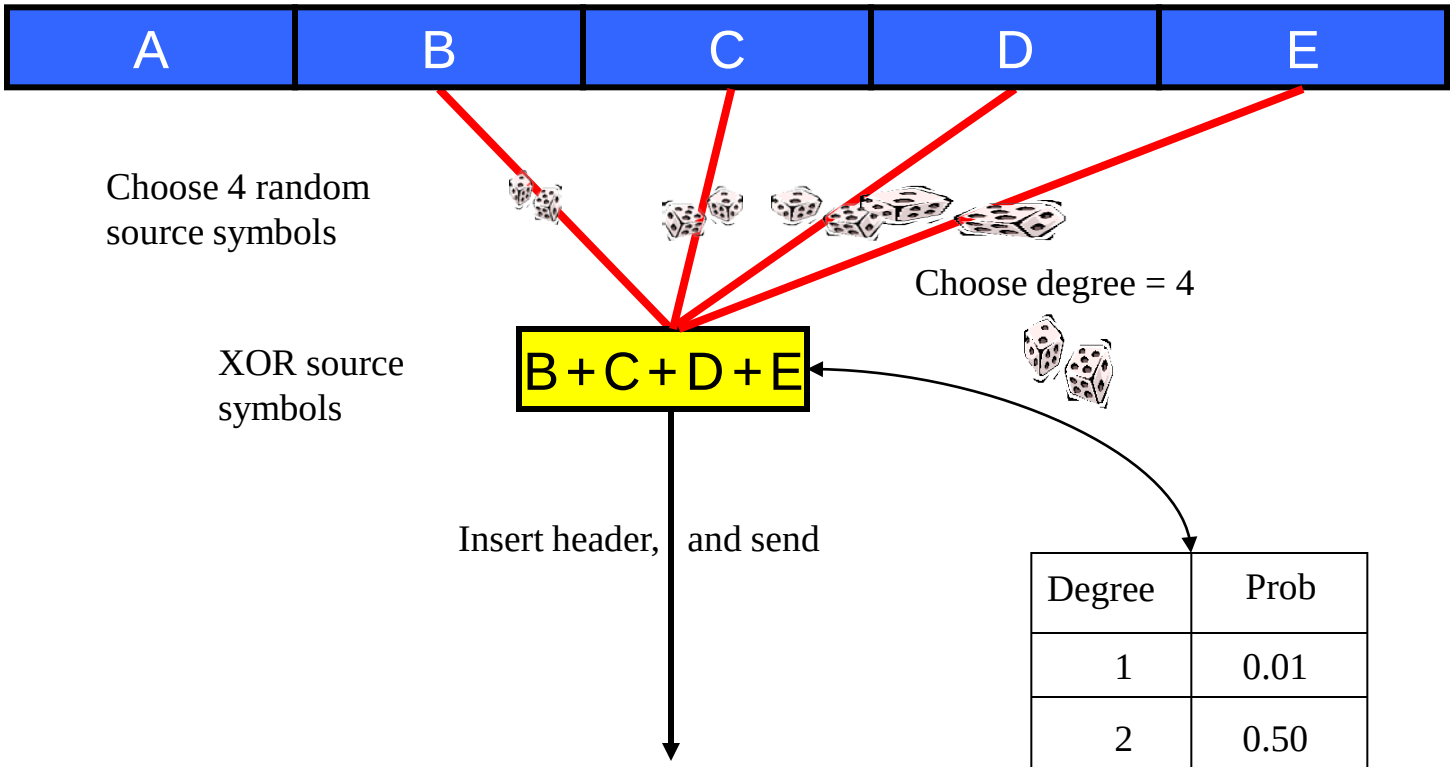
# LT encoding


A
B
C
D
E
Source block
Choose 2 random source symbols
Choose degree = 2
XOR source symbols
B + D
Insert header, and send
Degree
Prob
1
0.01
2
0.50
3
0.17
4
0.08
Degree Distribution

# LT encoding


A
B
C
D
E
Source block
Choose 1 random source symbol
Choose degree = 1
Copy source symbol
C
Insert header, and send
Degree
Prob
1
0.01
2
0.50
3
0.17
4
0.08
Degree Distribution

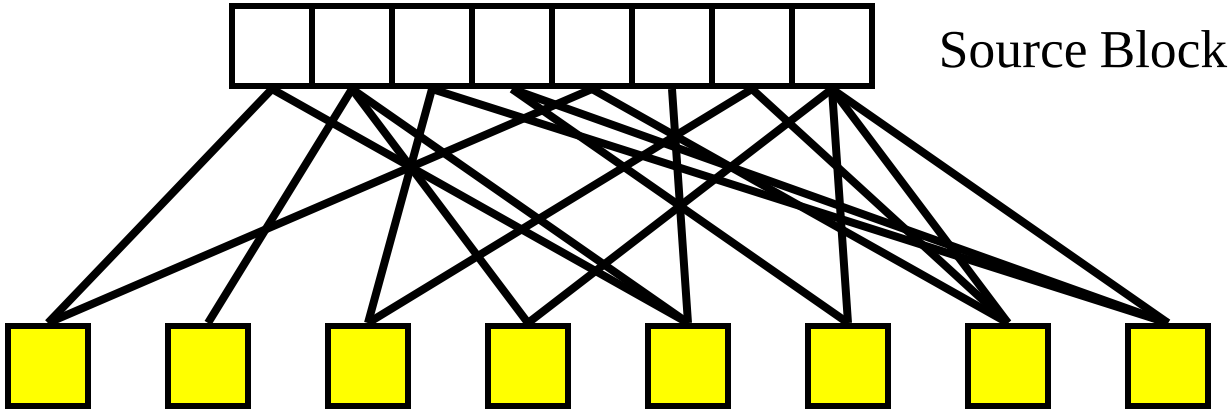
# LT encoding




| Degree | Prob |
|---|---|
| 1 | 0.01 |
| 2 | 0.50 |
| 3 | 0.17 |
| 4 | 0.08 |
| ⋮ | ⋮ |

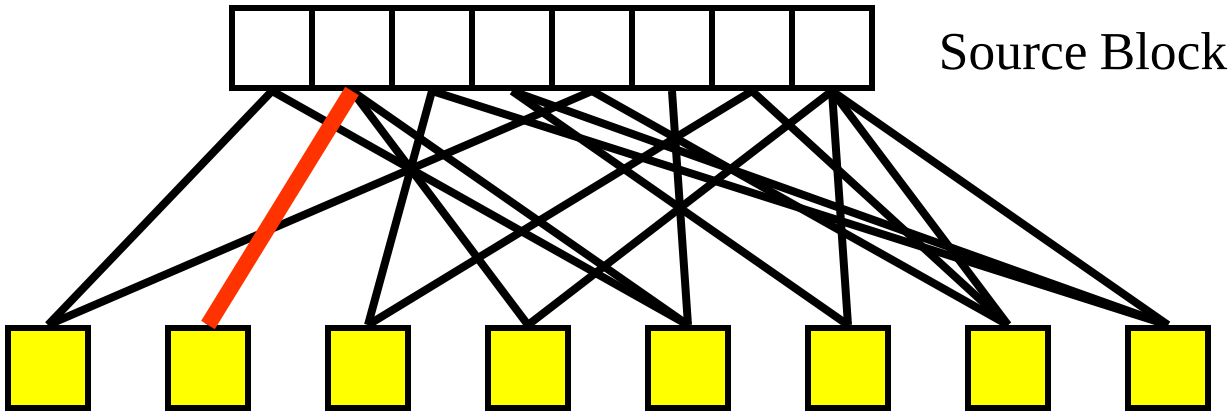
**Degree Distribution**

# Belief propagation decoding



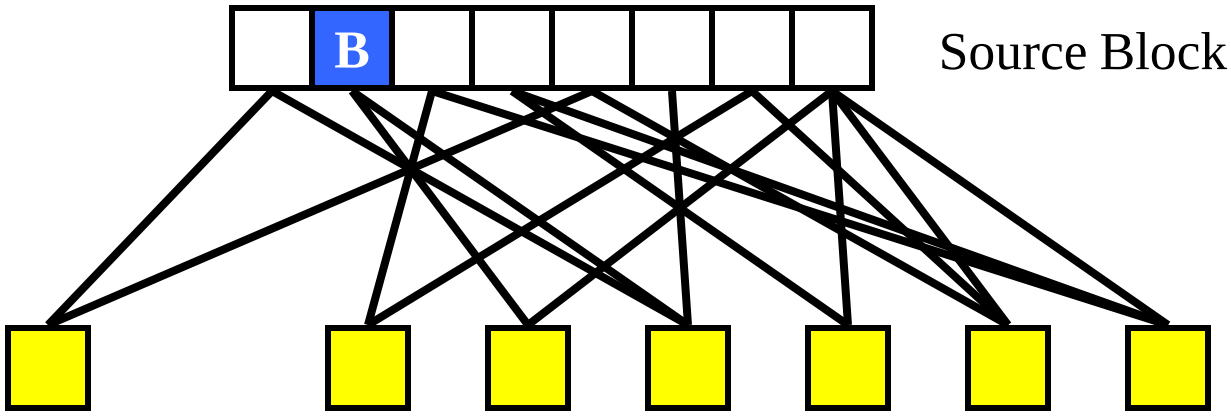

Collect enough encoded symbols and set up graph
between encoded symbols and source symbols to be decoded

# Belief propagation decoding



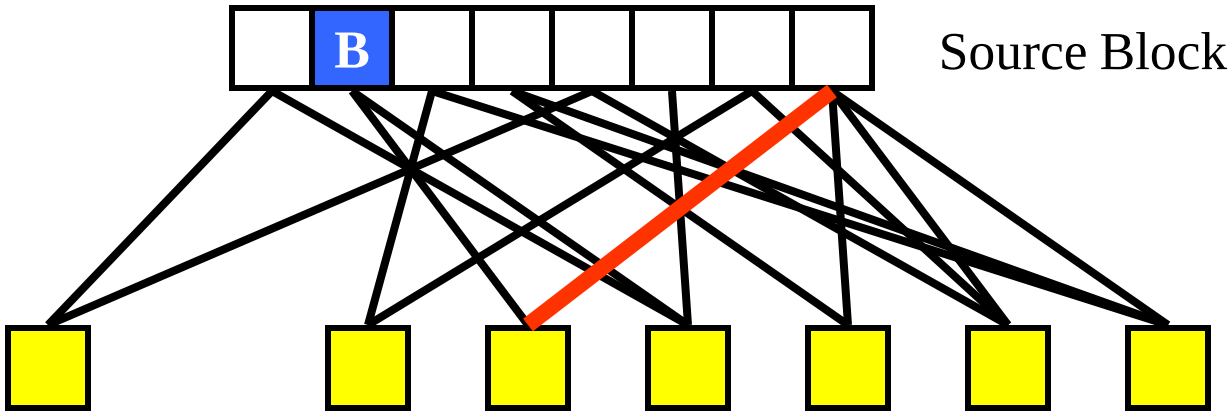

Identify encoded symbol with one unrecovered neighbor
STOP if none exists

# Belief propagation decoding



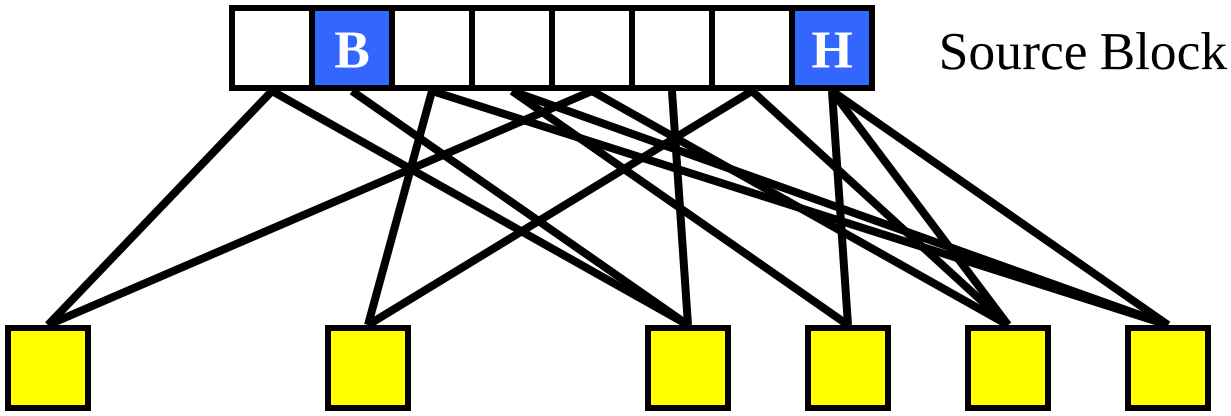

Unrecovered source symbol value is the value of all recovered neighbors XORed into the encoded symbol value

# Belief propagation decoding



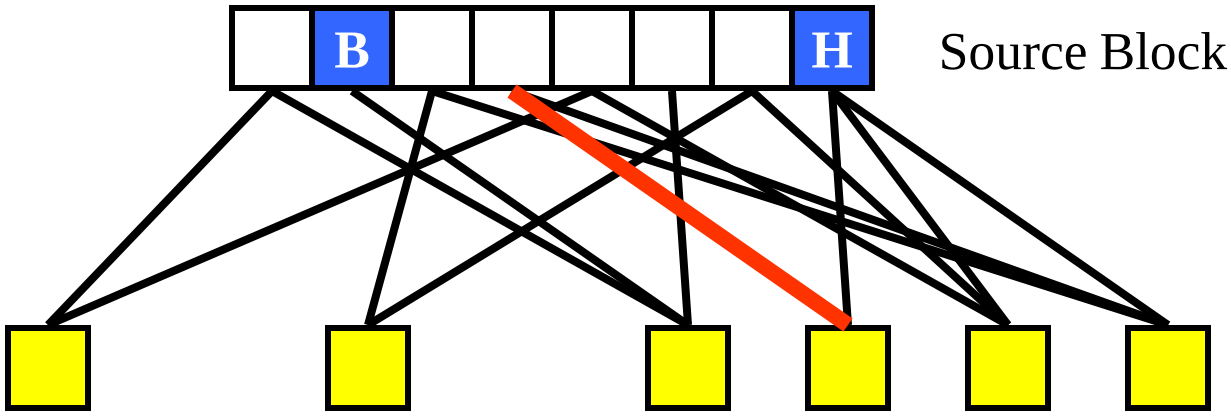

Identify encoded symbol with one unrecovered neighbor
STOP if none exists

# Belief propagation decoding



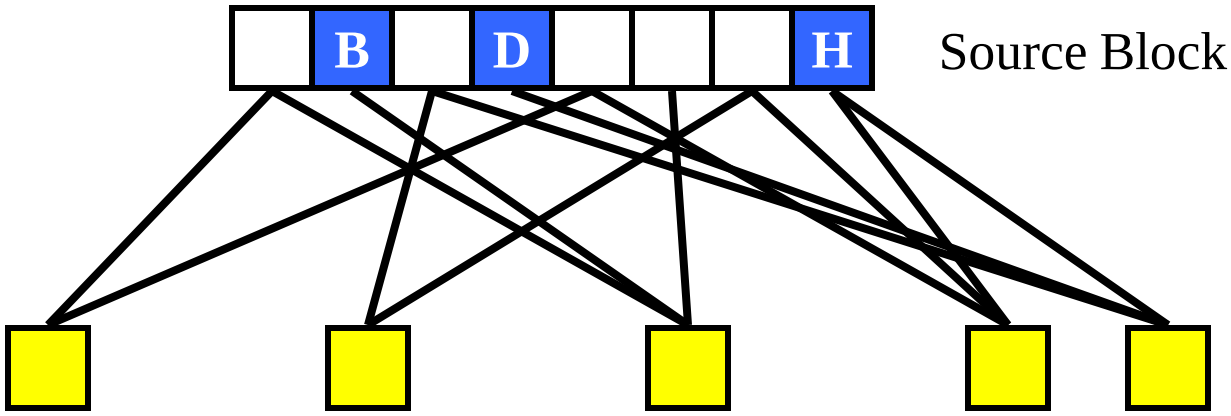

Unrecovered source symbol value is the value of all recovered neighbors XORed into the encoded symbol value

# Belief propagation decoding




Identify encoded symbol with one unrecovered neighbor
STOP if none exists

# Belief propagation decoding




Unrecovered source symbol value is the value of all recovered neighbors XORed into the encoded symbol value

# Belief propagation decoding

Source Block (recovered)

# Some technical ingredients of Raptor codes

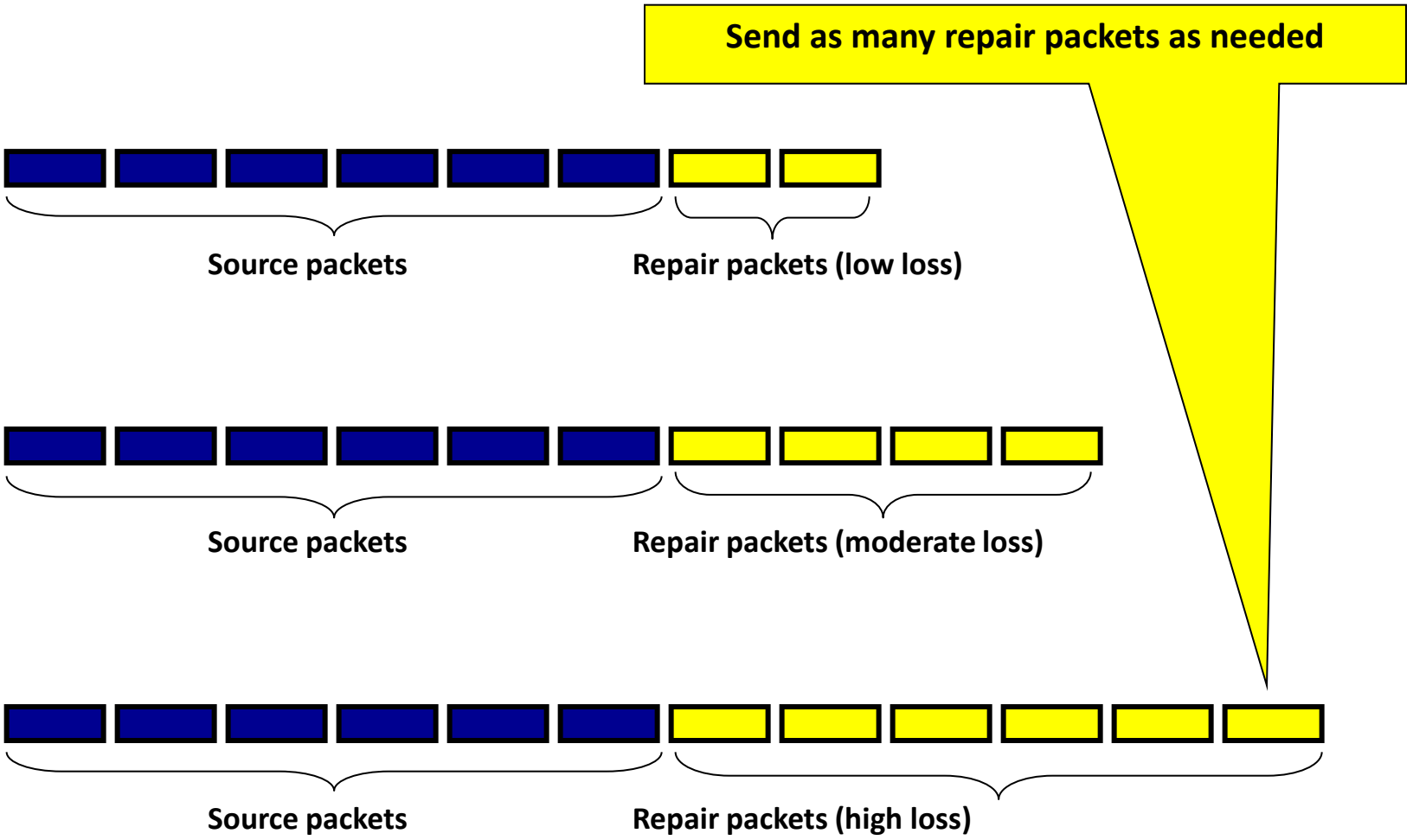
**Ingredient (first code that incorporated ingredient)**

- **LT code (Raptor RFC 5053)**
- Pre-coding (Raptor RFC 5053)
- Inactivation decoding (Raptor RFC 5053)
- Systematic construction (Raptor RFC 5053)
- Larger finite fields (RaptorQ RFC 6330)
- Permanent inactivations (RaptorQ RFC 6330)

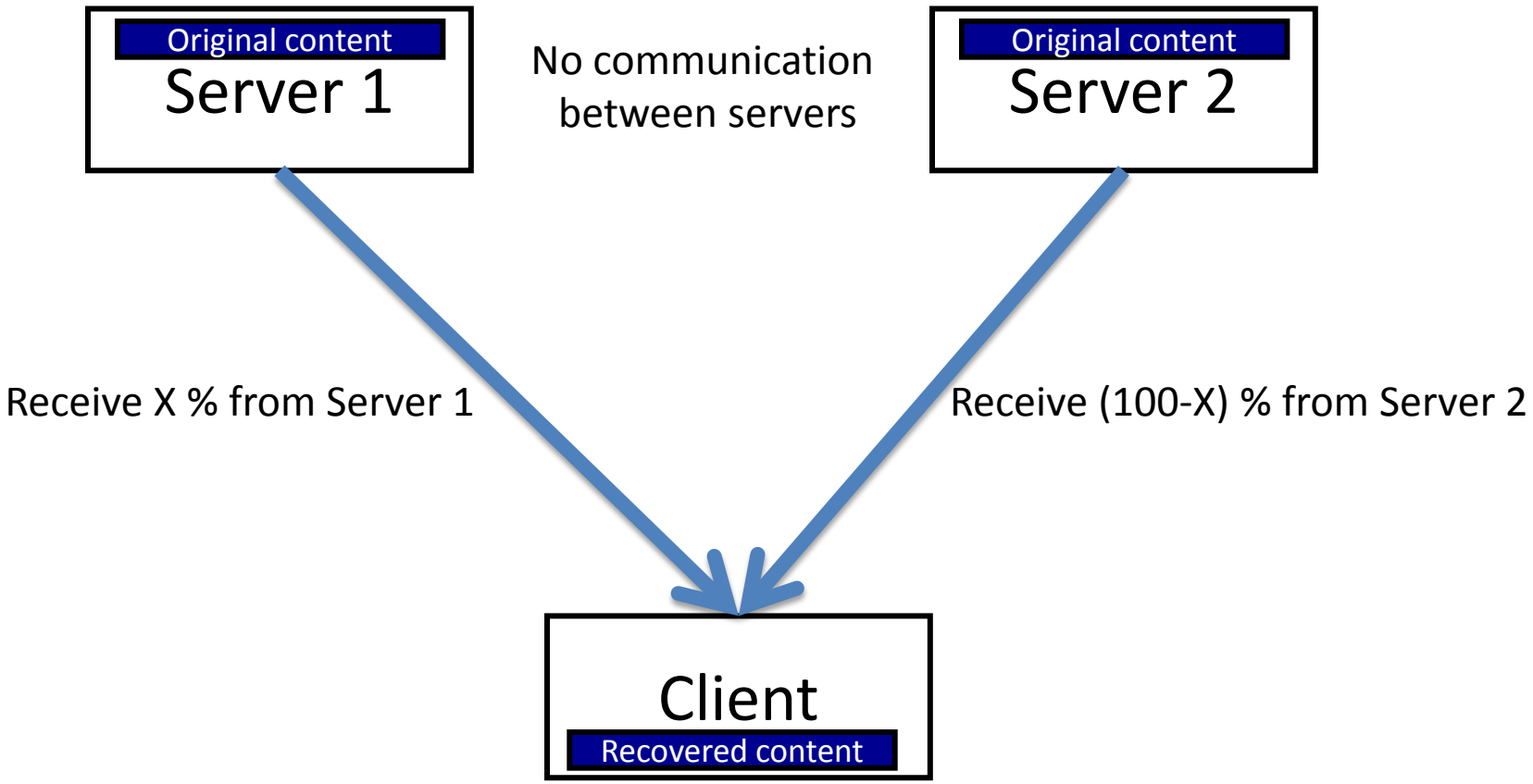
# Raptor codes key properties

- Dynamic packet loss protection
  - "Fountain" codes which means "rateless"
    - No limit to number of encoding symbols (n) that can be generated from any number of source symbols (k)
    - Unlimited amount of loss protection possible
  - No "pre provisioning required"; everything on-the-fly
- Exceptionally computationally efficient
  - Linear complexity (encoding and decoding)
  - Quadratic complexity for Reed-Solomon IETF RFC 5510
- Low transmission and reception overheads
  - Good overhead for Raptor IETF RFC 5053
  - Essentially zero overhead for RaptorQ IETF RFC 6330
  - High and variable overhead for LDPC IETF RFC 5170

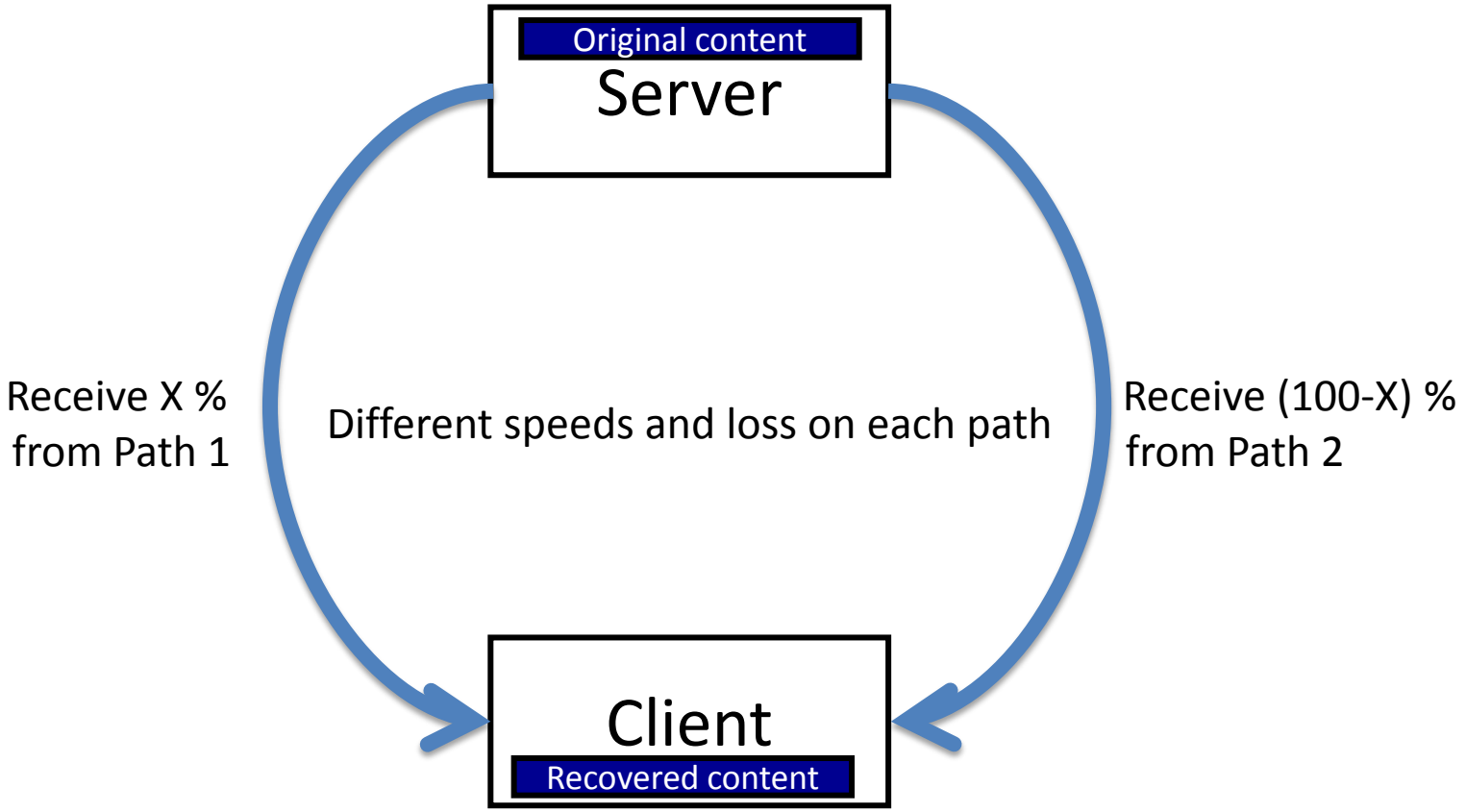
# Raptor codes fountain flexibility



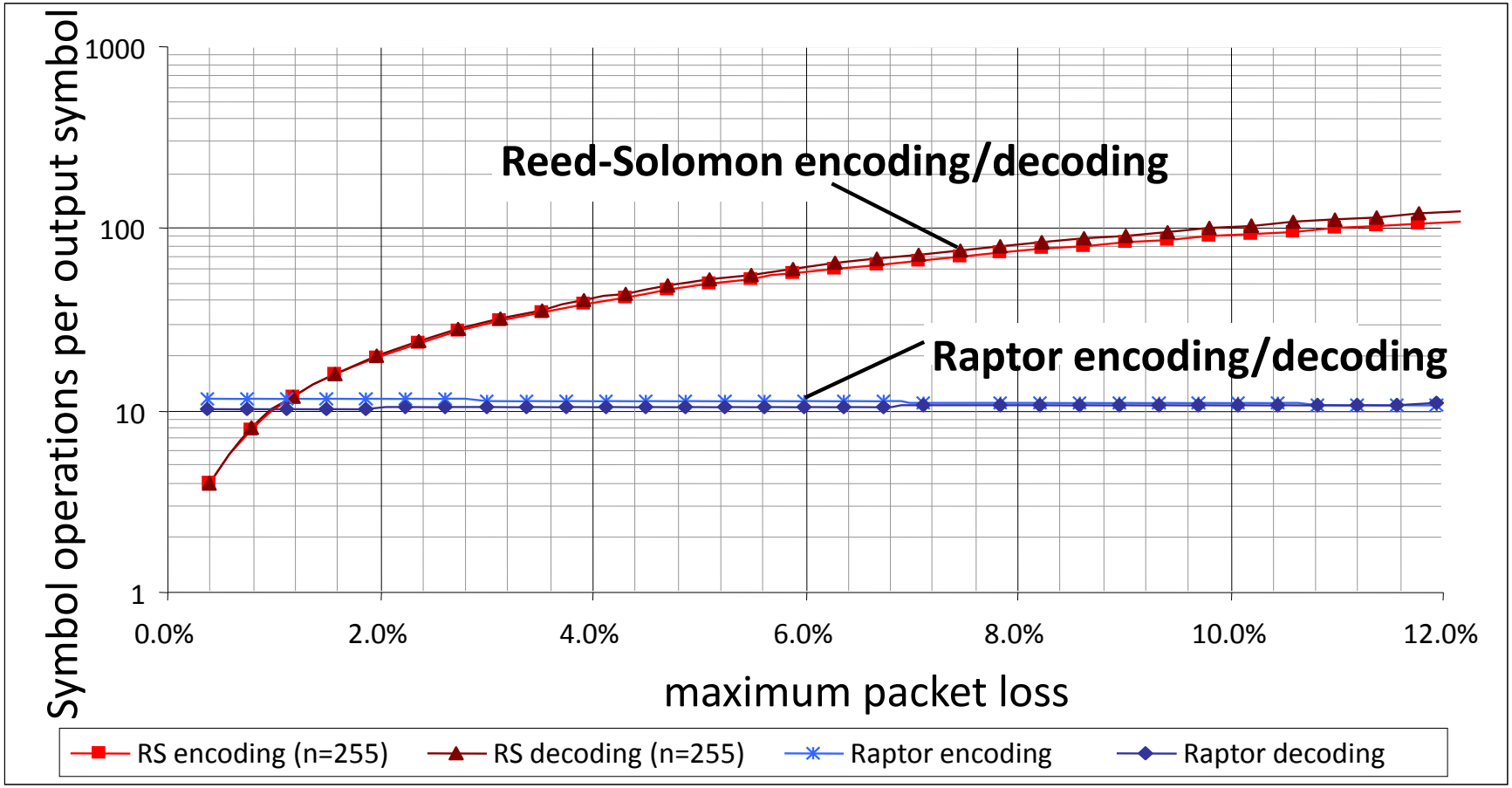
# Raptor codes fountain flexibility



# Raptor codes fountain flexibility

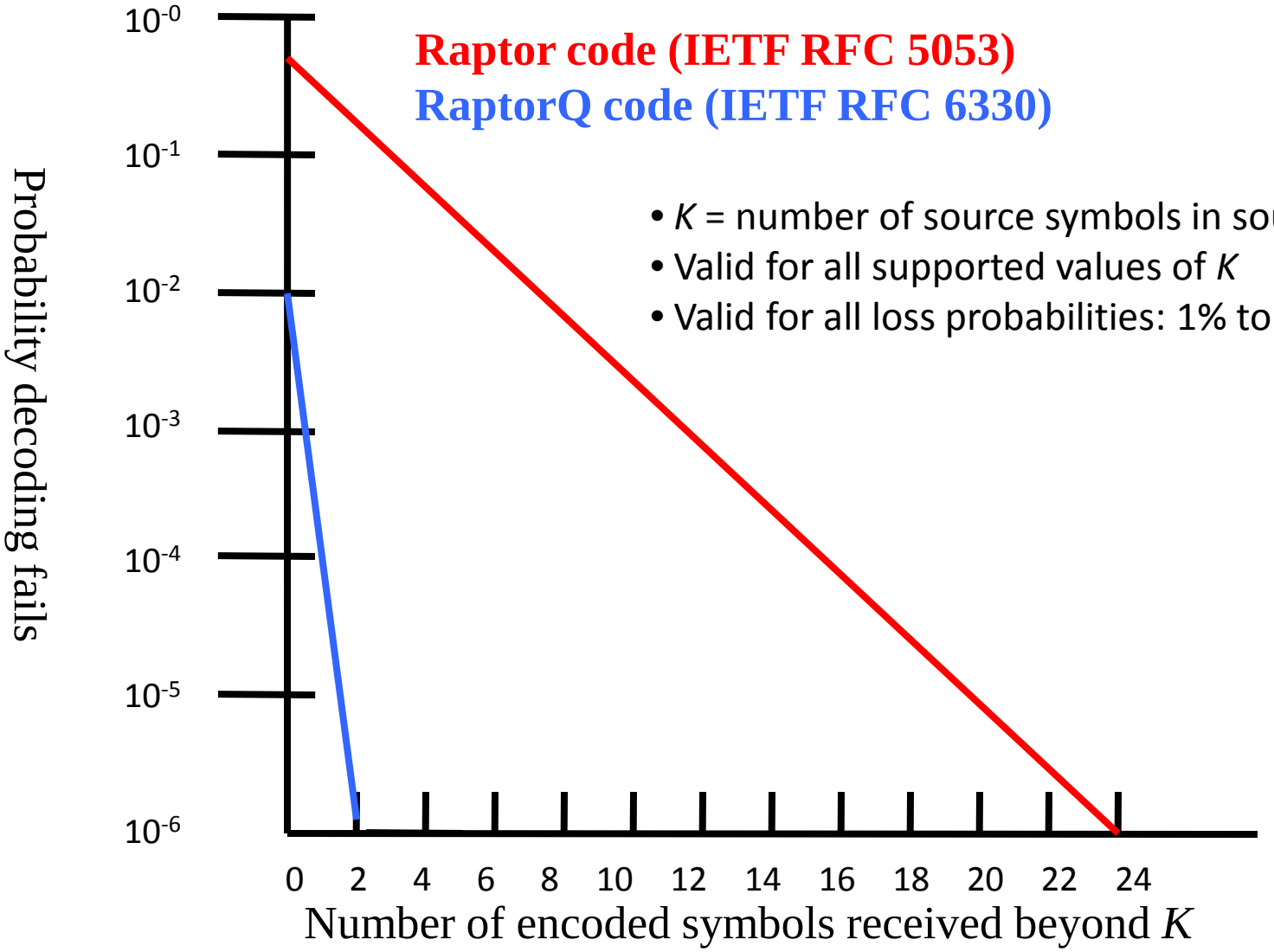


# Raptor codes computational complexity




***In comparison to other typical alternative FEC technologies, Raptor codes are an order of magnitude or more less complex***

# Raptor codes overhead



# How to deploy Raptor codes

- Raptor codes are versatile and simple to deploy
  - Application Layer FEC does not require special hardware
    - Raptor codes software library is incorporated into the application
    - Encoding and decoding speeds are more than sufficient for all applications
    - Same software library for file transfer or data/video streaming applications
    - Software development kit with well-defined and tested APIs
  - Point-to-point architecture
    - Encoder and Decoder pair
    - Same library for client/server, peer-to-peer, or broadcast deployments
  - Small footprint can be ported to many types of platforms
    - Encoder and Decoder libraries are less than 100KB each
    - Library does not perform any floating-point operations
    - No memory allocation or other platform specific functionality
    - Extremely portable to essentially any platform

# Why Raptor codes are used

- Values of Raptor:
  - Enhances user experience – video quality and smooth playback
  - Allows Content Providers to have consistent high QOS across services
  - Allows applications not otherwise possible due to network limitations
  - Speeds-up data delivery with variable network conditions
  - Increases overall system efficiency: less bandwidth, lower power, simpler architecture for application development
- Problems Raptor codes can address:
  - Provides high quality delivery even when packets are lost
  - Provides reliable delivery even when packets are lost
  - Provides more predictable delivery time when delivery is over multiple connections

# How organizations are using Raptor codes

- Content Delivery
  - Enterprise sends 30 GB+ database update weekly to 500 sites via satellite broadcast network
  - Military broadcasts image data via private mobile communications in foreign countries
  - Cellular operator broadcasts news flashes and video clips
  - Navigation service provider broadcasts map updates to fleets of vehicles
  - Digital cinema distribution over broadcast satellite
- Streaming
  - IPTV deployments in Asia and Europe on set-top boxes
  - Military airborne operations stream live video feeds to ground troops
  - Ground sensor networks transmit data to central command centers
  - Point-to-point video conferencing
  - On-scene digital news video feeds over multiple 3G connections

# Broadcasting content to mobile devices

Satellite or Terrestrial Delivery

Operations and Broadcast Center

Multimedia content, navigational content, etc.

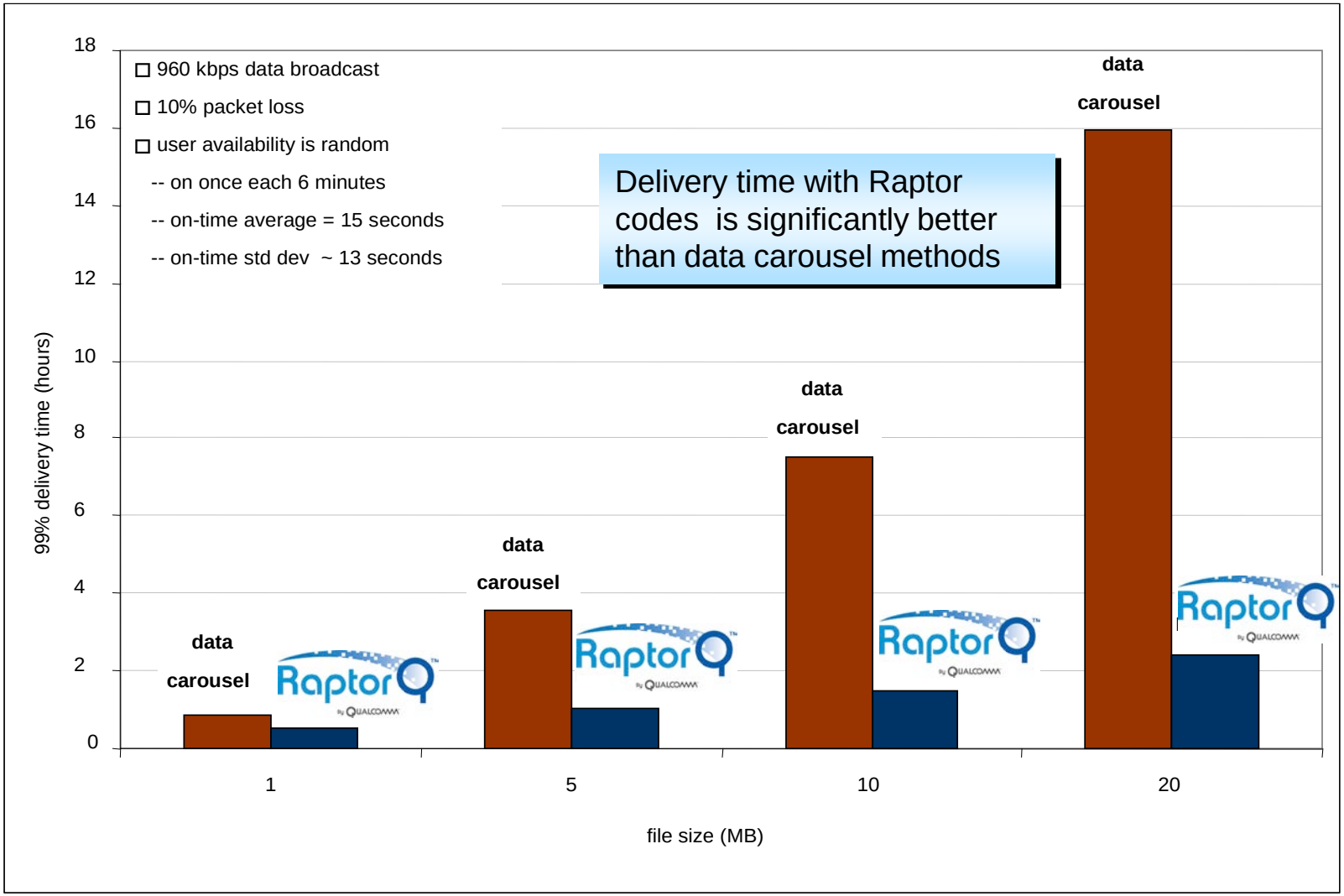
Entertainment, Communications & Navigation Systems

Many, many Receivers

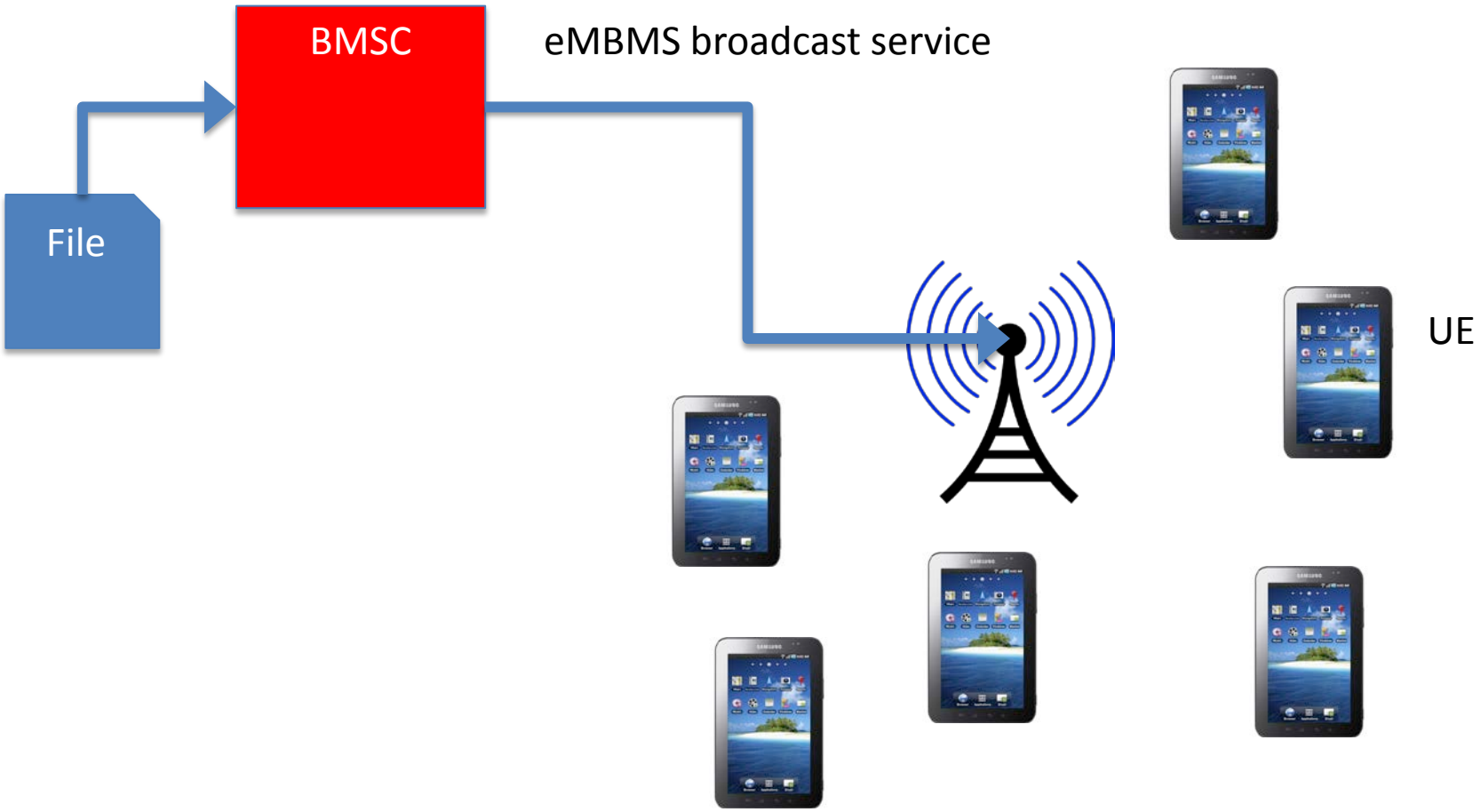
# Challenges of mobile broadcast

- Common transmission channel for all receivers
- No feedback from receivers
- Receivers are in various locations with varying conditions
- None of the dynamic unicast channel protocols apply
  - No adjustment of modulation parameters
  - No adjustment of error-correction scheme
  - No ARQ of radio packets
  - No transmission power control
- Application-layer unicast reliability protocols not applicable
  - No HTTP/FTP/TCP for file delivery
  - No HTTP/TCP for streaming
  - No RTP with retransmit for streaming

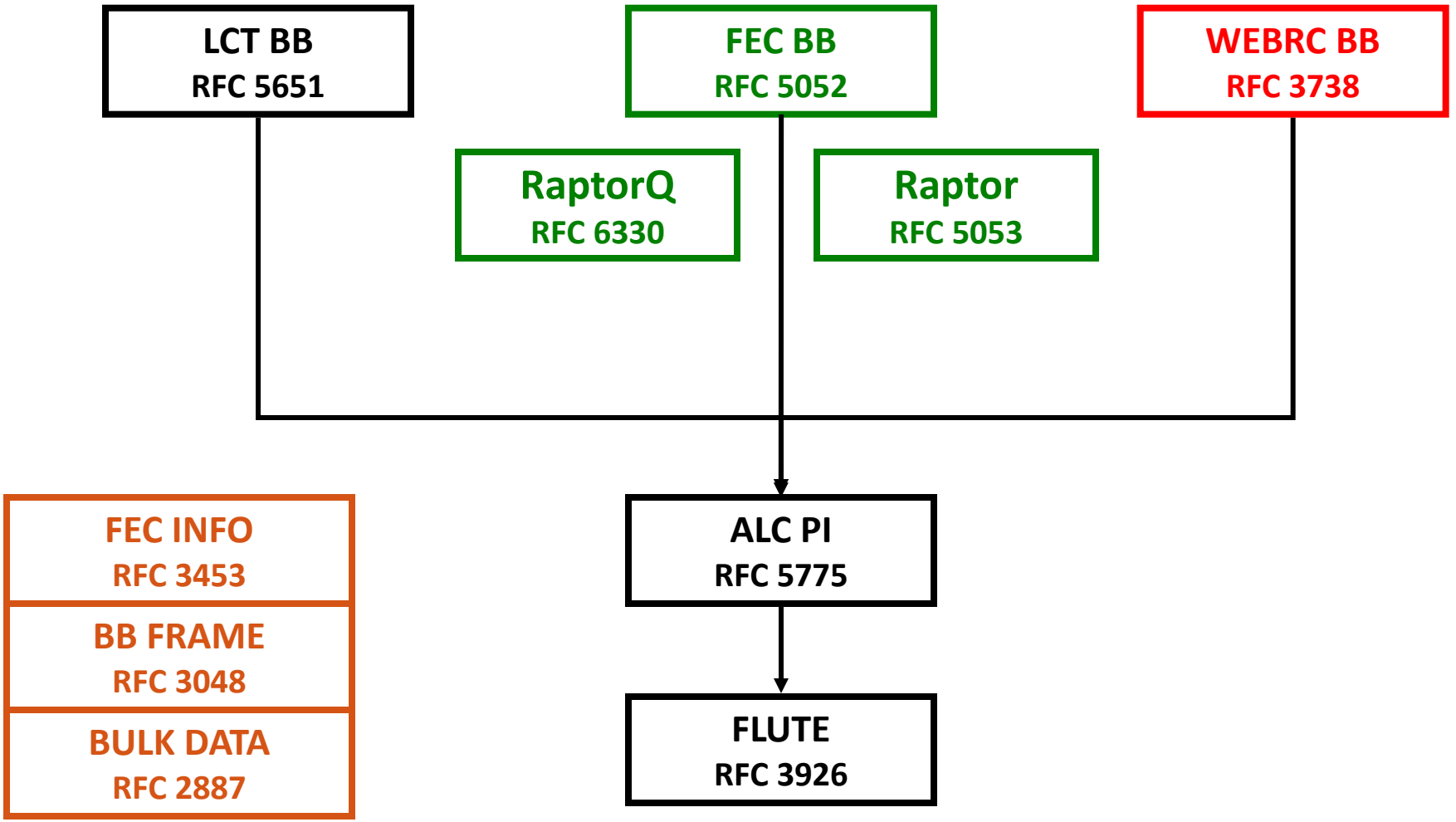
# Raptor codes versus traditional data carousel method



# 3GPP eMBMS – broadcast file delivery



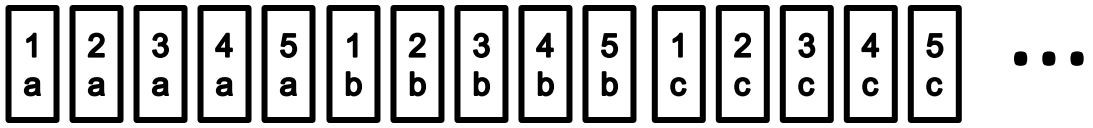
# IETF RMT Broadcast/Multicast File Delivery



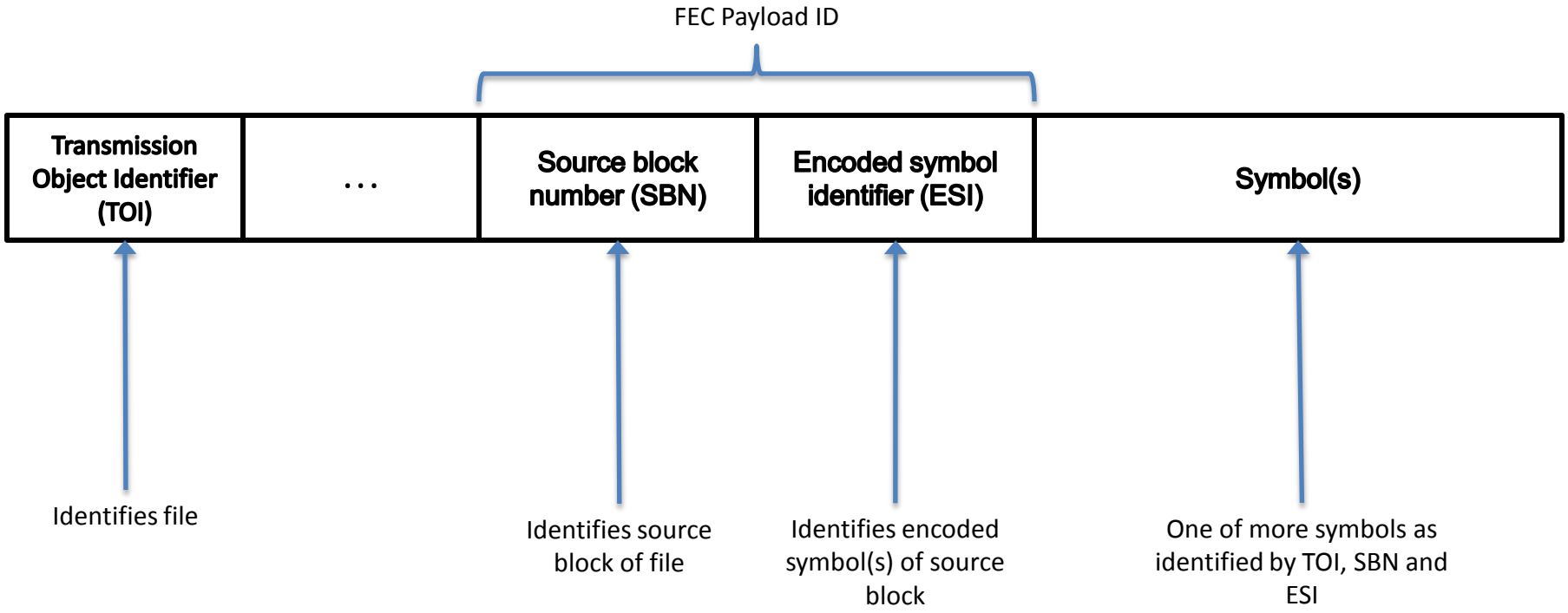
# FLUTE file delivery

Partition file into source blocks
FEC encode each source block independently

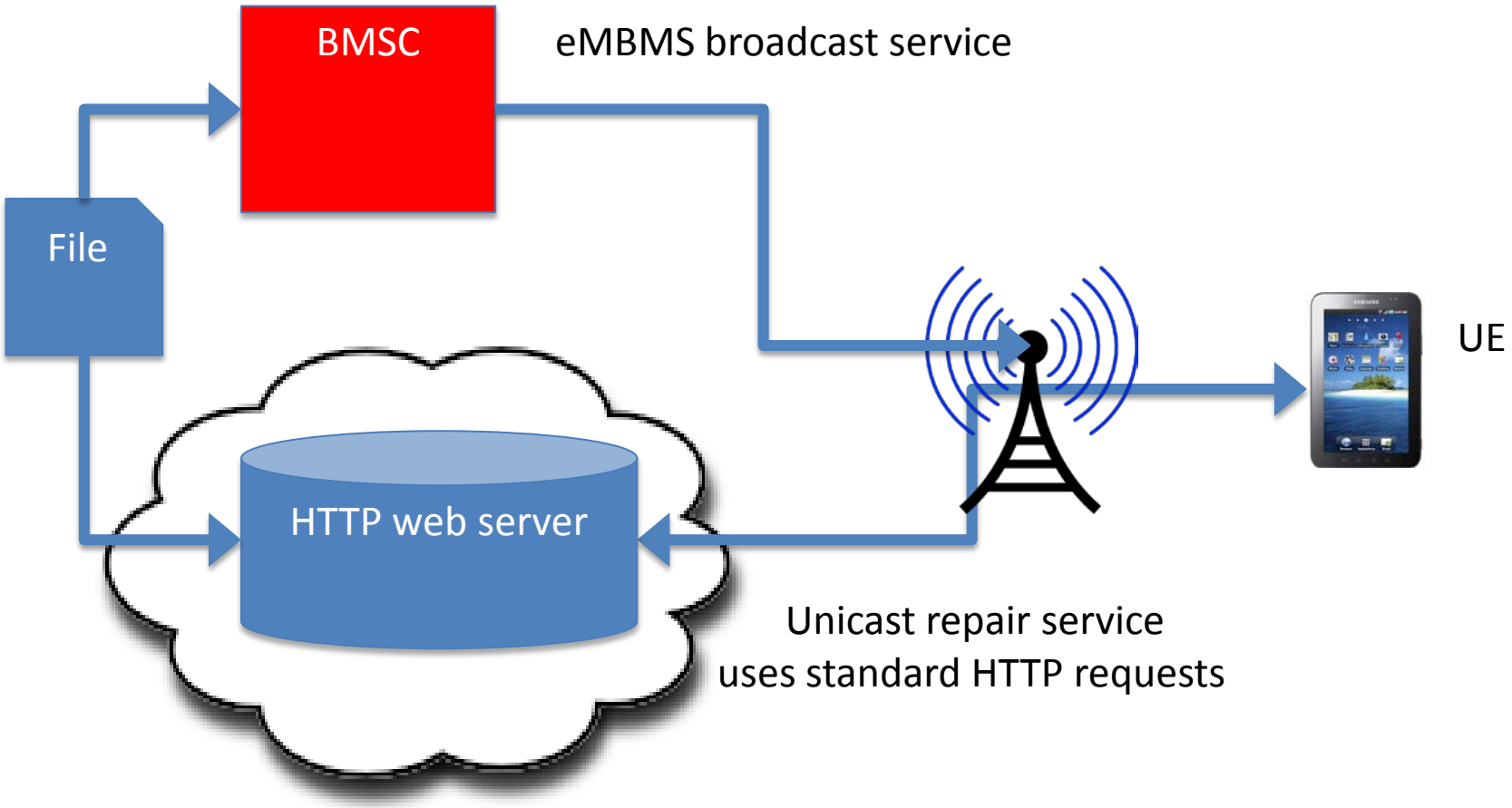
Place encoded symbols into FLUTE packets and transmit interleaved



# FLUTE packet format



# Proposed enhancement to 3GPP eMBMS



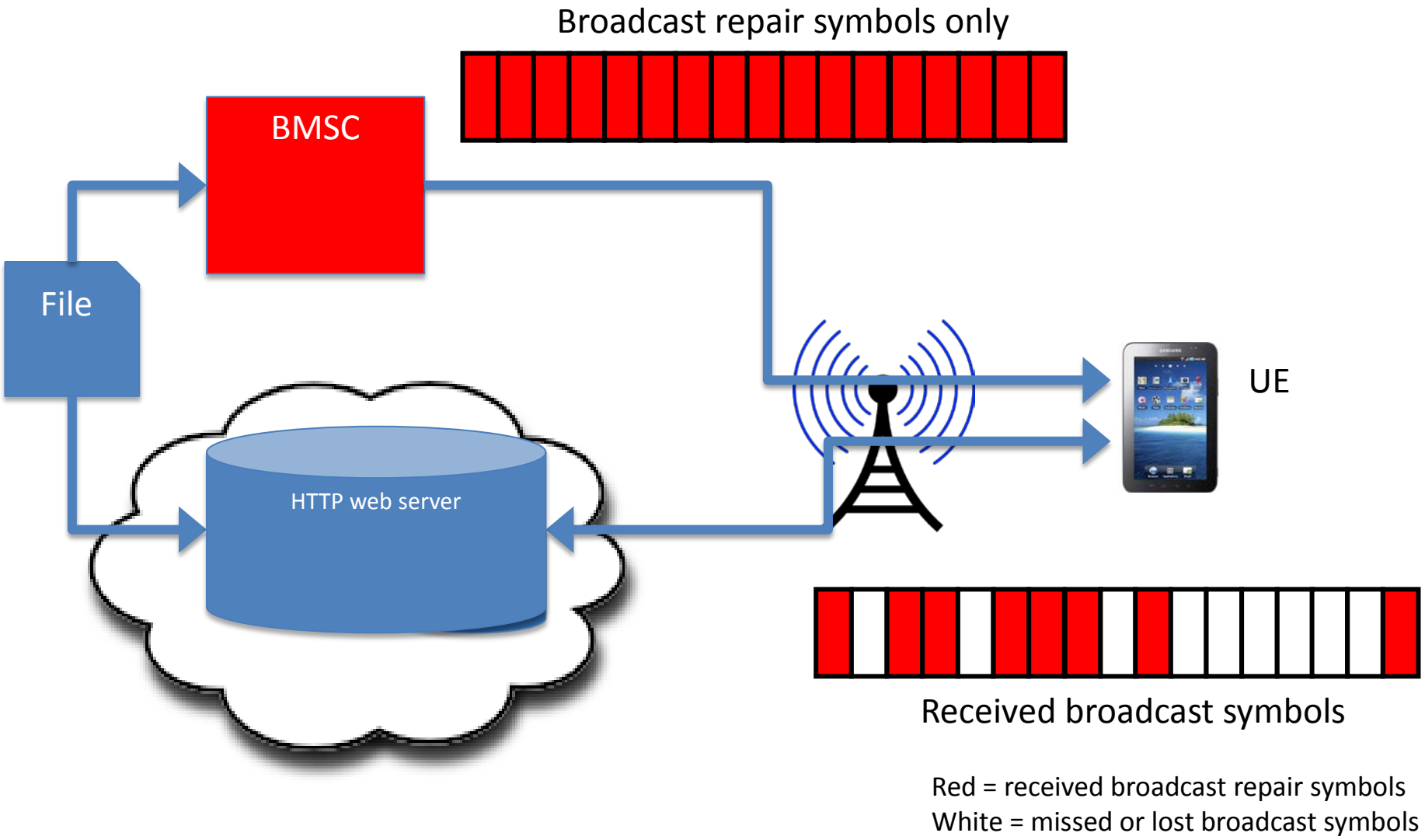
# Hybrid service deployment

- Use standard HTTP web servers as “repair” servers
  - UEs receive data for file from eMBMS broadcast session
  - UE can determine HTTP byte range requests for missing source symbols based on FEC OTI and received source symbols
  - UE makes requests to receive missing source symbols
  - Amount of HTTP data requested for each source block = source block size - amount of broadcast data received for that source block

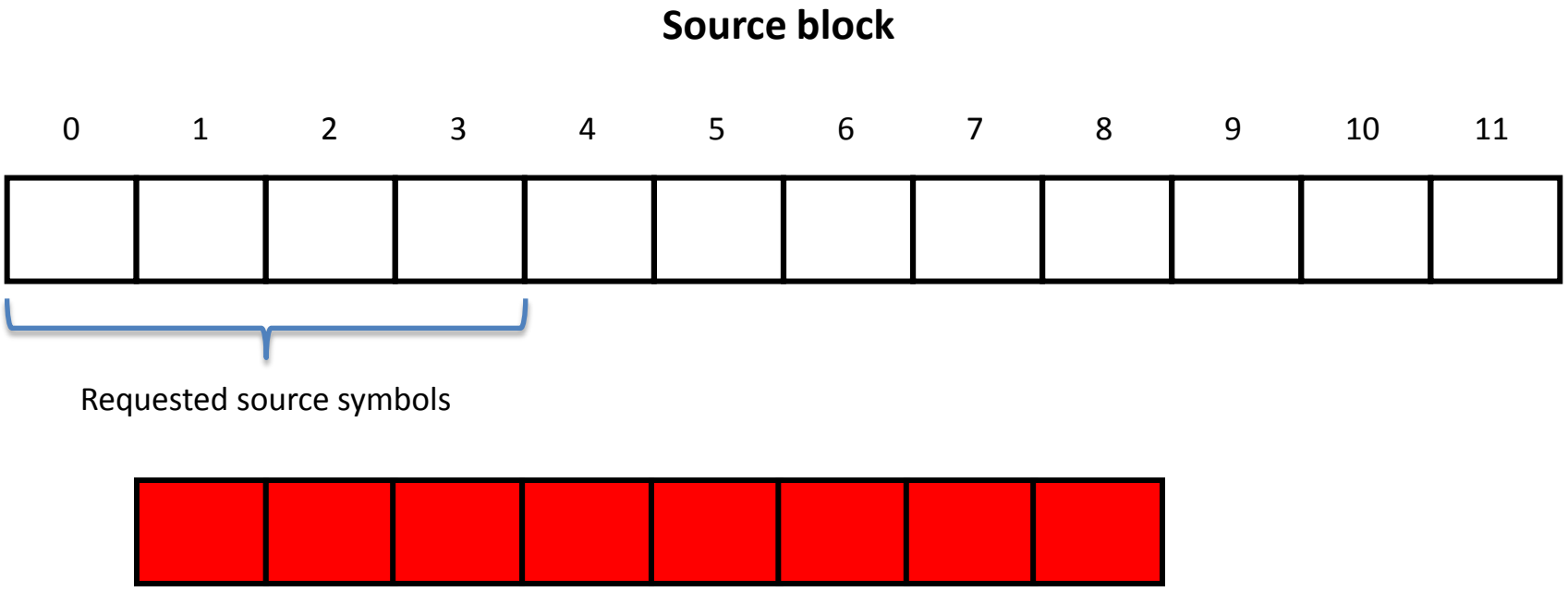
# What to broadcast?

- Broadcast only repair symbols – non-intuitive, but …
  - Simplifies HTTP byte range requests to initial prefixes of blocks
  - Few HTTP byte range requests required for unicast repair
    - independent of loss patterns
  - High caching efficiency
    - each UE will have the same request pattern
  - Easy to deploy hybrid HTTP/broadcast use cases
  - Raptor/RaptorQ decoding very efficient
    - resource usage dominated by non-FEC decoding operations such as data movement between RAM and flash memory

# Broadcast only repair symbols



# File with one source block and no sub-blocks
# Broadcast only repair symbols




All source symbols missing
Request 4 source symbols 0-3

Blue = received broadcast source symbols
Red = received broadcast repair symbols
White = missed or lost broadcast symbols

# Example

- File size 20 MB
- Packet payload size = 1,024 bytes
- Encoded as 20 source blocks
  - Source block size = 1 MB
  - 1,024 source symbols per source block
  - 20,480 source symbols in the file
- Suppose UE receives 19,000 packets from the broadcast
  - UE needs to request 1,480 source symbols
  - This is approximately 1.5 MB of unicast repair data

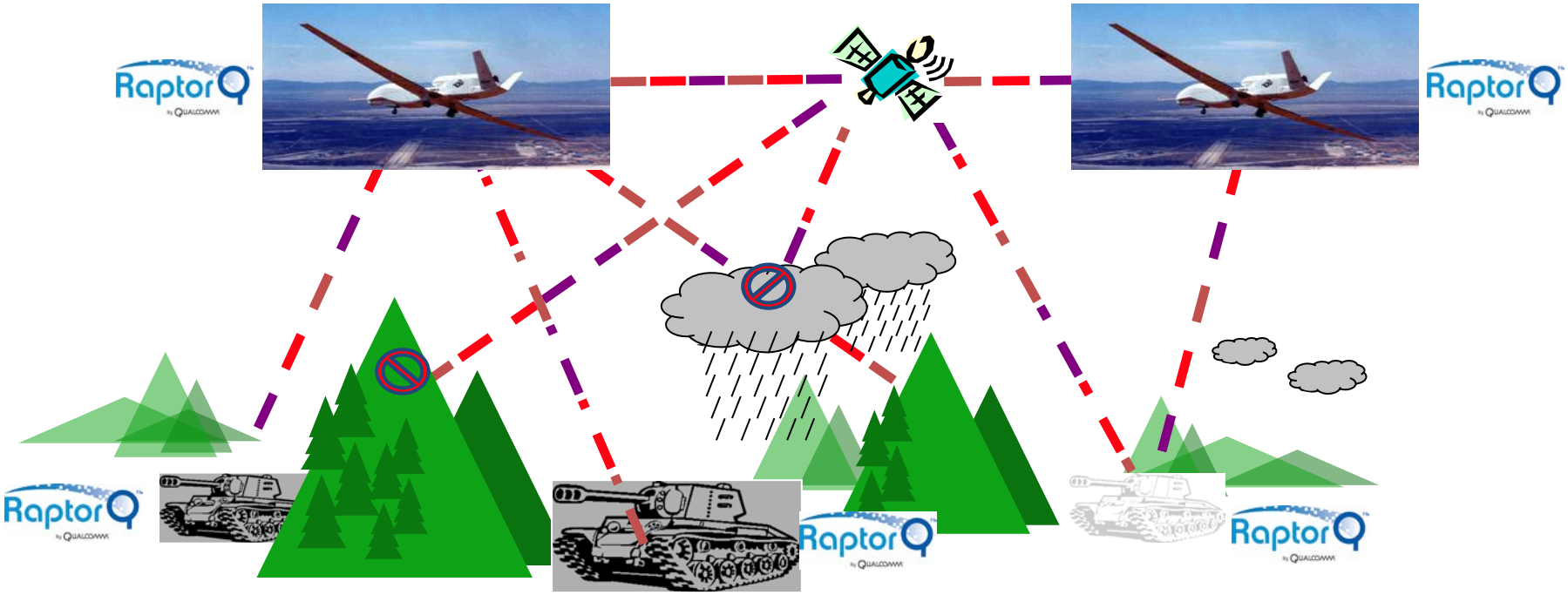
# Unicast repair requests

- Low HTTP request and response overhead & simple request pattern
  - Number of requests = # source blocks independent of loss and pattern of loss
  - 20 HTTP byte range requests for prefixes of the 20 source blocks
- High caching efficiency
  - Different UEs make completely overlapping requests
  - Pattern of requests from different UEs independent of individual UE loss patterns

# Summary of approach and benefits

- Hybrid Internet and eMBMS services
  - Allows using standard HTTP web servers as repair servers
  - Allows advanced hybrid Internet/eMBMS services not otherwise possible
    - Allows eMBMS to be used as a “traffic offload service” for Internet content
    - Allows Internet infrastructure to provide eMBMS unicast repair functionality

# Defense Communications



## Hostile and challenging conditions

- Intermittent connectivity – blocking or loss of any particular source
- Stealth receivers – unidirectional reliability
- Ad-hoc network – moving transmitters and receivers
- Multilink – reception of different packets over different links
- Rapidly and widely varying network conditions

# High QOS Video over Wired or Wireless: Entertainment, Conferencing or Surveillance

over Imperfect Networks
IP Network over FTTP, DSL, or Wireless

<u>Unprotected Video</u>

- Compression compounds impact of errors
- Block errors
- Buffering
- Video and Audio loss

<u>Raptor Protected</u>

- Steady & Crisp Audio & Video
- Loss can exceed 15% or more
- Low latency
- Low processing overhead

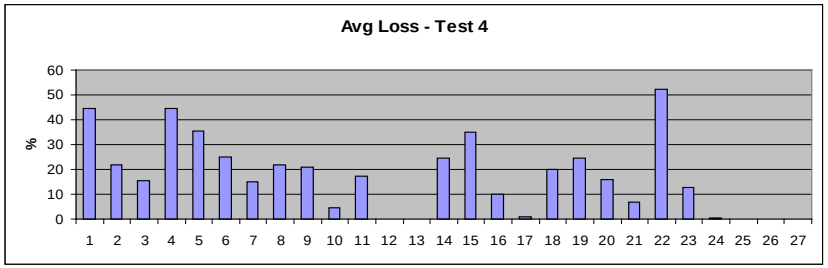
over Imperfect Networks
IP Network over FTTP, DSL, or Wireless

Airborne Networking Example

# Video Streaming Over an Airborne Network

No protection

25% Raptor codes protection




***Robust Protection of Streaming Data!***

# Raptor codes benefits and advantages

- Efficiency
  - Dramatic link margin improvement
  - Exceptionally fast encoding and decoding algorithms (scalable)
  - Error-free delivery over any data network and recovering lost data packets without requiring re-transmission from the sender
  - Good combination of PHY-FEC and AL-FEC optimizes overall efficiency
- Flexible - operates ideally in a huge range of network conditions
  - Multi-link/multi-network architecture
  - High/unknown/variable loss
  - Large latencies
  - Intermittent connectivity
  - Broadcast channels with no back channel
- Superior mobile receiver operations
  - Implemented in host software without any special hardware support
  - Consistent low CPU decoding complexity
  - Suitable for low cost and low power consumer devices (mobile phones, handheld devices, etc.)
  - Provides longer battery life
- See www.qualcomm.com/raptorq for more information!

RaptorQ™ By Qualcomm

# Thank you!