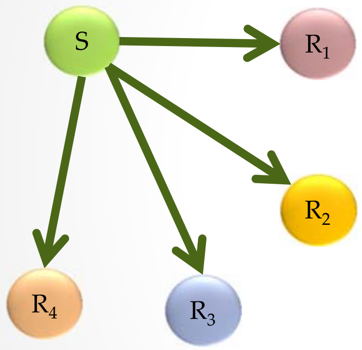
Stopping a Rapid Tornado with a Puff

José Lopes and **Nuno Neves**

Email: nuno@di.fc.ul.pt

LASIGE, University of Lisboa

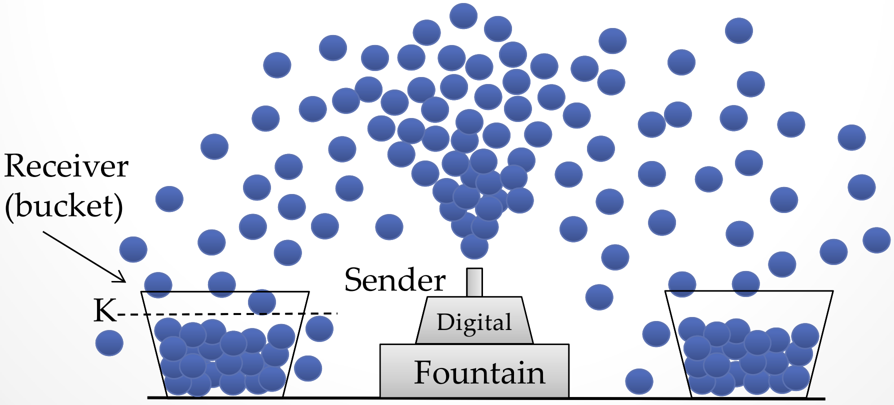
Example: Point-to-multipoint communication



- For a large number of receivers TCP does not scale
  - every receiver requires a separate data stream
  - sender needs to keep track of what arrives at each receiver
- UDP can be used
  - scales effortlessly
  - best effort: loss rate ↑ ⇒ degraded experience
- Difficult to provide a scalable broadcast service on the Internet
  - ***it would be interesting to have reliability whilst retaining UDP's efficiency***

FEC and fountain codes

- Forward Error Correction (FEC)
  - split data into symbols (e.g., packets)
  - encode symbols in a way that introduces redundancy capable of recovering missing symbols
- Fountain codes
  - endless supply of encoded symbols
  - recover original data with **any** *K* encoded symbols (with high probability)

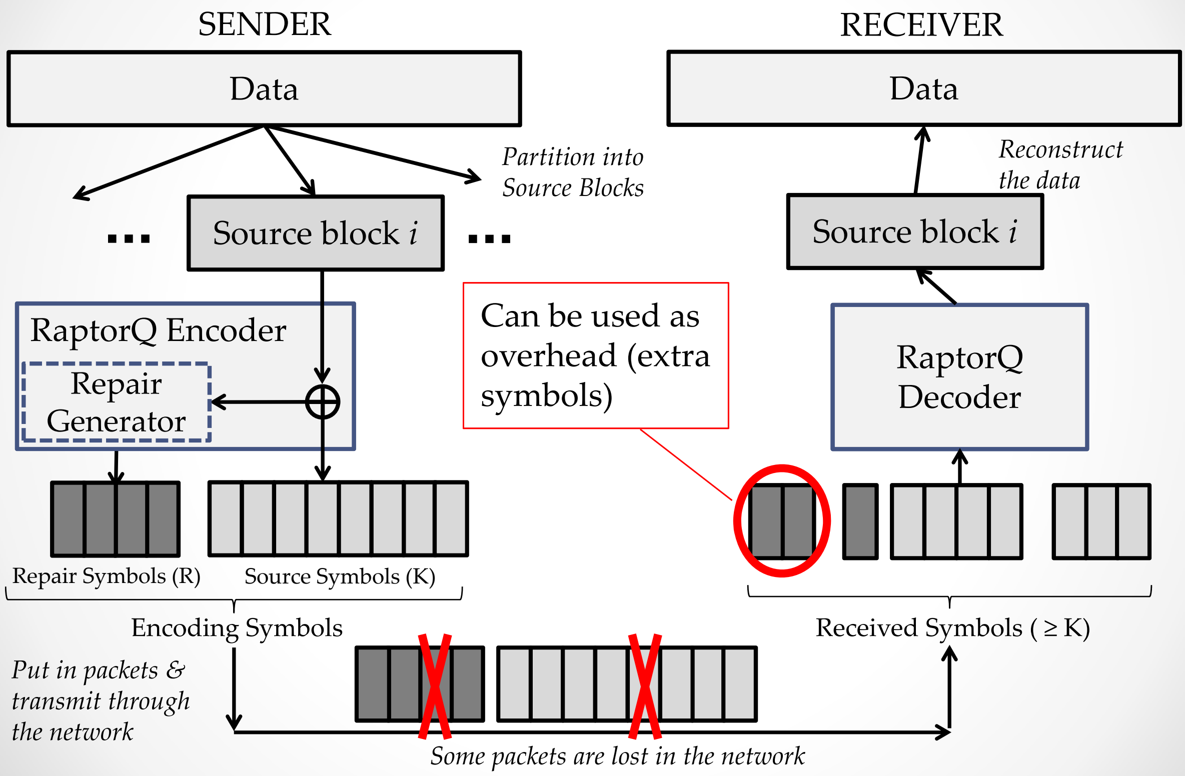


Rapid Tornado Codes

- Raptor codes are the most recent fountain codes
- Their secret lies in applying a "pre-code" to the source symbols, before encoding
  - which reduces complexity to O(1) (per-symbol)
- RaptorQ is their flagship
  - efficient encoding/decoding ⇒ permanent inactivation
  - steeper overhead-failure curve ⇒ non-binary alphabets
  - standardized as IETF RFC 6330
- *Systematic* code ⇒

  encoded symbols = source + repair symbols

Transmission overview using RaptorQ



SENDER

Data

Partition into Source Blocks

… Source block *i* …

RaptorQ Encoder

Repair Generator

Repair Symbols (R)

Source Symbols (K)

Encoding Symbols

Put in packets & transmit through the network

Some packets are lost in the network

RECEIVER

Data

Reconstruct the data

Source block *i*

RaptorQ Decoder

Can be used as overhead (extra symbols)

Received Symbols (≥ K)

Decoding failure probabilities

- Incredibly low probabilities of failure against accidental faults

	K (number of source symbols)								
	0 Overhead $[\cdot 10^{-3}]$			1 Overhead $[\cdot 10^{-5}]$			2 Overhead $[\cdot 10^{-7}]$		
Loss	**10**	**26**	**101**	**10**	**26**	**101**	**10**	**26**	**101**
10%	0	5.4	5.7	0	0	3.8	0	0	2.5
20%	0	4.0	4.8	0	2.3	2.4	0	0	0.5
50%	0	3.9	4.9	0	1.6	2.5	0	0.9	1.2
60%	4.8	4.1	4.9	0	1.5	2.2	0	0	2.1
85%	0	12.7	4.7	0	0.8	2.4	0	0	1.3

NOTE: *overhead* number of received encoding symbols more than K ;
experiments were run between 20 to 30 million times for each setting

Can we stop RaptorQ with a puff?

Successful attack: through malicious faults force a decoding failure (of a source block)

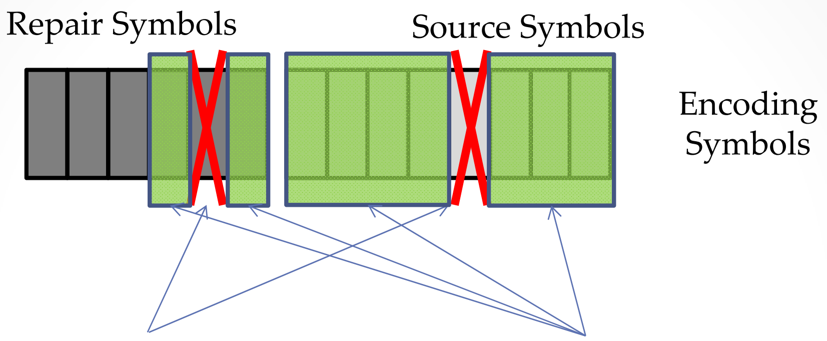
Rationale behind the attack

- Assume an attacker on the network that attempts to prevent decoding
- The attacker can create erasures on specific packets of the network
- Instead of randomly picking the encoding symbols, she/he cleverly chooses which packets may or may not reach the receiver

Objective:

1. How **big** of an **impact** can the attacker have?
2. Can the attack be done in a stealth way?

Rational behind the attack (2)



Attacker:
picks these encoding symbols to erase

Receiver :
for overhead = 1, the first K+1 encoding symbols are used to attempt decoding

We want to have 100% impact while minimizing the erasures remaining stealth

Over \ K	10	26	32	42	55	62	75
0	3	3	2	2	2	2	2
1	7	4	5	2	4	3	4
2	12	9	7	10	5	5	5
Over \ K	**84**	**91**	**101**	**153**	**200**	**248**	**301**
0	2	1	2	2	1	2	3
1	6	8	7	3	8	4	2
2	7	4	9	4	10	11	15
Over \ K	**355**	**405**	**453**	**511**	**549**	**600**	**648**
0	2	2	1	1	1	1	1
1	2	8	2	7	2	4	2
2	10		14	50	5		
Over \ K	**703**	**747**	**802**	**845**	**903**	**950**	**1002**
0	1	1	1	1	2	1	1
1	3	8	6	3	2	6	4
2	7			57			

Thank you! Any questions?

This was:

Stopping a Rapid Tornado with a Puff

José Lopes and Nuno Neves