Adaptive Multiaccess Streaming Codes

Muhammad Ahmad Kaleem
Gustavo Kasper Facenda
Ashish Khisti

University of Toronto

August 3, 2023

Table of Contents

Streaming

- Modern applications e.g. video conferencing, cloud computing, online gaming require low latency communication
- Transmission has to be robust to packet erasures
- Usual Coding Problem: A source encodes a (large) message to send over a noisy channel
- Streaming Problem: Source encodes a continuous sequence of messages under delay constraints

Low Latency Streaming

- Classical methods of error correction face challenges under latency requirements
  - Automatic Repeat Request (ARQ) has a large round trip delay
  - Standard Forward Error Correction (FEC) codes e.g. LDPC have large block lengths $\rightarrow$ high latency
- Streaming codes are FEC codes where symbols are decoded under a maximum delay
  - Common techniques include using systematic MDS codes, diagonal interleaving

Background

Single-Link (Point-to-Point) Streaming Codes

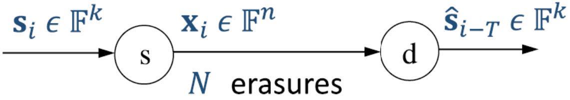
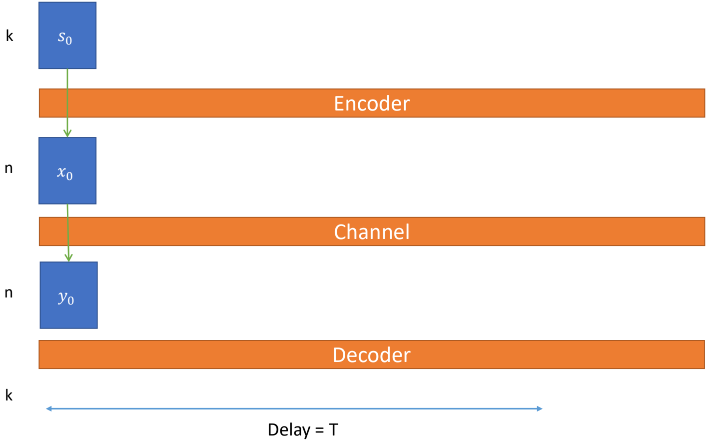


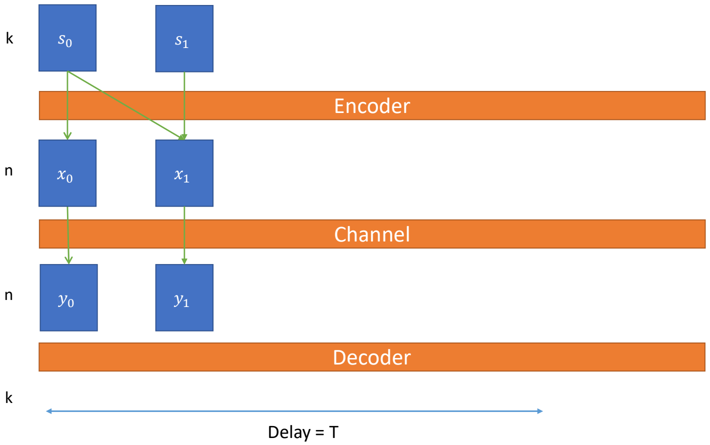
Figure: Single-Link Network (Fong et al., 2019)

- Source wants to send a sequence of message packets to destination with a maximum delay of T per message packet
- Adversarial erasure channel: maximum N erasures in window of size $T+1$
- Define a (n, k, T) streaming code. At each time $i \in \mathbb{Z}^+$
  - Source generates message packet $s_i \in \mathbb{F}^k$
  - Messages encoded into packet $x_i \in \mathbb{F}^n$ causally, transmitted over link: $x_i = f_i(s_0, \ldots, s_i)$
  - Destination receives $y_i \in \mathbb{F}^n$ equal to x_i or * (erasure)
  - Destination generates estimate $\hat{s}_{i-T}$ (delay T) as a function of $\{y_\tau\}_{\tau=0}^{i}$
- Rate $R = \frac{k}{n}$, capacity $C(T, N) = \frac{T+1-N}{T+1}$

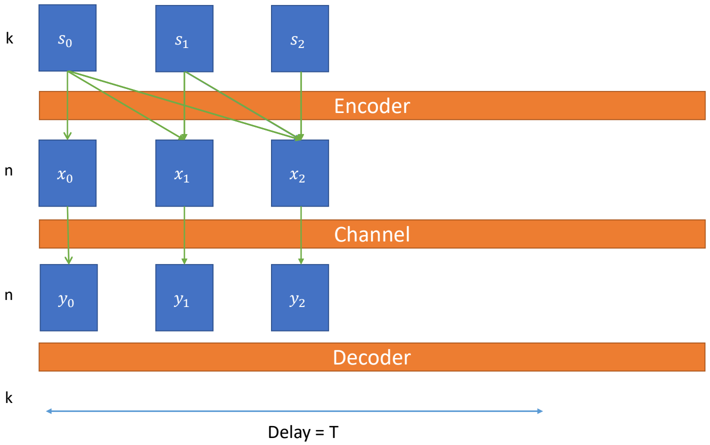
Single-Link Streaming Codes (cont'd)



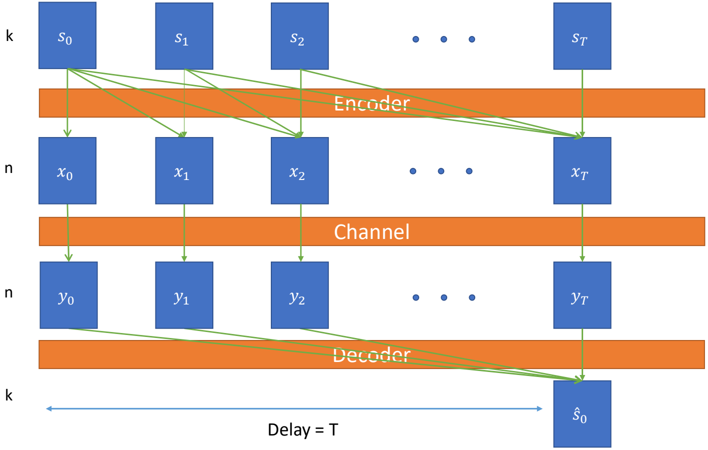
Single-Link Streaming Codes (cont'd)



Single-Link Streaming Codes (cont'd)



Single-Link Streaming Codes (cont'd)



Example

Time i	0	1	2	3	4
$s_i[0]$	$s_0[0]$	$s_1[0]$	$s_2[0]$	$s_3[0]$	$s_4[0]$
$s_i[1]$	$s_0[1]$	$s_1[1]$	$s_2[1]$	$s_3[1]$	$s_4[1]$
$s_{i-2}[0] + s_{i-1}[1]$	0	$s_0[1]$	$s_0[0] + s_1[1]$	$s_1[0] + s_2[1]$	$s_2[0] + s_3[1]$

(a) Symbols transmitted by the source from time 0 to 4.

Time i	0	1	2	3	4	5
$\hat{s}_{i-2}[0]$	0	0	$\hat{s}_0[0]$	$\hat{s}_1[0]$	$\hat{s}_2[0]$	$\hat{s}_3[0]$
$\hat{s}_{i-1}[1]$	0	$\hat{s}_0[1]$	$\hat{s}_1[1]$	$\hat{s}_2[1]$	$\hat{s}_3[1]$	$\hat{s}_4[1]$

(b) Estimates constructed by the destination from time 0 to 5.

Figure: Point-to-Point code with $T = 2, N = 1, n = 3, k = 2$

Three Node Relay Network

Figure: Three Node Network (Fong et al., 2019)

- Source sends messages to destination via relay
- 2 links, maximum N_1 erasures in first, N_2 in second
- Total delay T
- Define a (n_1, n_2, k, T) streaming code and $R = \min(\frac{k}{n_1}, \frac{k}{n_2}) = \frac{k}{\max(n_1, n_2)}$

Symbol-wise Decode-and-Forward (SWDF)

- Relay decodes source *symbols*, then encodes them and sends to destination
- Consider $T = 5, N_1 = 2, N_2 = 3$

Table: Source to Relay Code. $(3, 1)$ MDS code.

Time i	0	1	2	3	4	5	6
a_i	a_0	a_1	a_2	a_3	a_4	a_5	a_6
a_{i-1}		a_0	a_1	a_2	a_3	a_4	a_5
a_{i-2}			a_0	a_1	a_2	a_3	a_4

Table: Relay to Destination Code. $(4, 1)$ MDS code.

Time i	0	1	2	3	4	5	6	7
a_{i-2}			a_0	a_1	a_2	a_3	a_4	a_5
a_{i-3}				a_0	a_1	a_2	a_3	a_4
a_{i-4}					a_0	a_1	a_2	a_3
a_{i-5}						a_0	a_1	a_2

Adaptive Relaying

Adaptive Relaying

- SWDF construction assumes pessimistic scenario where first link packets always suffer N_1 erasures
- However, at most N_1 erasures in an interval of length $T+1 \rightarrow$ not all packets will have N_1 erasures

Adaptive Relaying (cont'd)

Table: Source to Relay Code

Time i	0	1	2	3	4	5
a_i	a_0	a_1	a_2	a_3	a_4	a_5
a_{i-1}		a_0	a_1	a_2	a_3	a_4
a_{i-2}			a_0	a_1	a_2	a_3

Table: Relay to Destination Code.

Time i	0	1	2	3	4	5	6	7
a_{i-2}			a_0	a_1	a_2	a_3	a_4	a_5
a_{i-3}				a_0	a_1	a_2	a_3	a_4
a_{i-4}					a_0	a_1	a_2	a_3
a_{i-5}						a_0	a_1	a_2

- Idea: Observe erasures from source
- Consider time i, assume burst of length p affecting source packet $(0 \leq p \leq N_1)$.
- Relay starts to receive symbols from message packet i at time $i + p$
- Relay can start transmitting at $i + p$ instead of $i + N_1$

Adaptive Relaying (Example)

Table: Source to Relay Code

Time i	0	1	2	3	4	5
$a_i[1:6]$	a_0	a_1	a_2	a_3	a_4	a_5
$a_{i-1}[1:6]$		a_0	a_1	a_2	a_3	a_4
$a_{i-2}[1:6]$			a_0	a_1	a_2	a_3

Table: Relay to Destination Adaptive Code

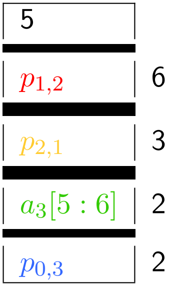
Time i	0	1	2	3	4	5	6	7	
$a_{i-2}[1:6]$				$a_1[1:6]$					6
$p_{i-3,1}[1:6]$					$p_{1,1}$				6
$p_{i-4,2}[1:6]$						$p_{1,2}$			6
$p_{i-5,3}[1:6]$							$p_{1,3}$		6
$a_{i-1}[1:3]$				$a_2[1:3]$					3
$a_{i-2}[4:6]$					$a_2[4:6]$				3
$p_{i-3,1}[1:3]$						$p_{2,1}$			3
$p_{i-4,2}[1:3]$							$p_{2,2}$		3
$p_{i-5,3}[1:3]$								$p_{2,3}$	3
$a_i[1:2]$	$a_0[1:2]$			$a_3[1:2]$					2
$a_{i-1}[3:4]$		$a_0[3:4]$			$a_3[3:4]$				2
$a_{i-2}[5:6]$			$a_0[5:6]$			$a_3[5:6]$			2
$p_{i-3,1}[1:2]$				$p_{0,1}$			$p_{3,1}$		2
$p_{i-4,2}[1:2]$					$p_{0,2}$			$p_{3,2}$	2
$p_{i-5,3}[1:2]$						$p_{0,3}$			2

Adaptive Relaying (cont'd)

- Previously: symbols transmitted per timeslot $= \frac{k}{T+1-N_1-N_2} = \frac{6}{1} = 6$
- Now: $\frac{k}{T+1-p-N_2} = 3, 2 < 6$
- Allows for value of n_2 to be reduced $\rightarrow \frac{k}{n_2}$ increases
- Scheme achieves higher rates than SWDF for $N_2 > N_1$ and equal rates otherwise (Facenda et al., 2023b)
- Main issue: Very high values of k (message length) to ensure divisibility by $T + 1 - N_2, \ldots, T + 1 - N_1 - N_2$: $k = \Pi_{i=0}^{N_1}(T + 1 - N_2 - i)$

Worst Case Erasure Sequence

- n_2 changes depending on observed erasure sequence
- Want to find the worst case value of n_2
- Consider relay packet at time i, $x_i^{(r)}$
  - Contains source symbols from time interval $[i - T : i]$
- Find contribution of each time to this packet
- Formulate optimization problem



Subset Adaptation

- Idea: Instead of changing rates for all values $\in \{0, \ldots, N_1\}$, choose a subset $\mathcal{N} = \{j, N_1\}$, $0 \leq j < N_1$
- If packet from time i has $\leq j$ erasures, start transmitting at time $i + j$
- If more than j erasures, start transmitting at time $i + N_1$ (as in nonadaptive)
- Choose value of j which gives highest $\frac{k}{n_2}$

Example

- Consider same example as before and $j = 1$.

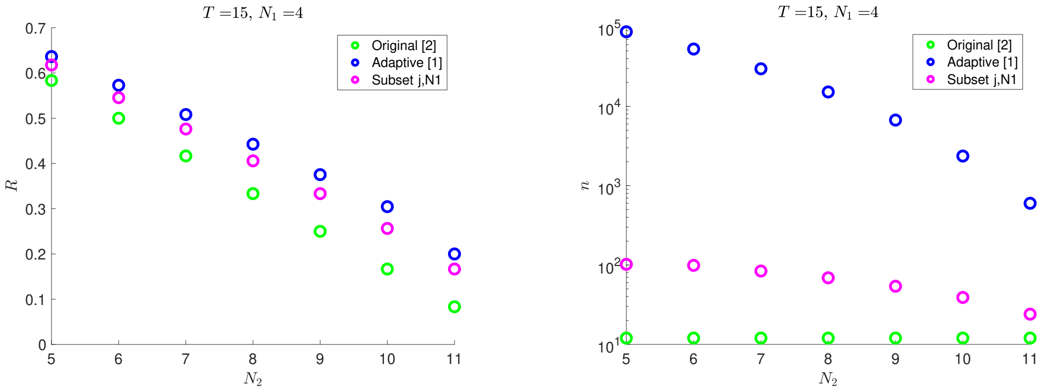
Table: Source to Relay Code

Time i	0	1	2	3	4	5
$a_i[1:6]$	a_0	a_1	a_2	a_3	a_4	a_5
$a_{i-1}[1:6]$		a_0	a_1	a_2	a_3	a_4
$a_{i-2}[1:6]$			a_0	a_1	a_2	a_3

Table: Relay to Destination Adaptive Code

Time i	0	1	2	3	4	5	6	7	
$a_{i-2}[1:6]$				$a_1[1:6]$					6
$p_{i-3,1}[1:6]$					$p_{1,1}$				6
$p_{i-4,2}[1:6]$						$p_{1,2}$			6
$p_{i-5,3}[1:6]$							$p_{1,3}$		6
$a_{i-1}[1:3]$		$a_0[1:3]$		$a_2[1:3]$	$a_3[1:3]$				3
$a_{i-2}[4:6]$			$a_0[4:6]$		$a_2[4:6]$	$a_3[4:6]$			3
$p_{i-3,1}[1:3]$				$p_{0,1}$		$p_{2,1}$	$p_{3,1}$		3
$p_{i-4,2}[1:3]$					$p_{0,2}$		$p_{2,2}$	$p_{3,2}$	3
$p_{i-5,3}[1:3]$						$p_{0,3}$		$p_{2,3}$	3

Performance



(a) Rate (b) Block Length

- Lower rates but more practical scheme due to lower k, n

Multiaccess Streaming

Multiaccess Networks

- Two sources transmitting messages to destination via relay
- Define a $(n_1, n_2, n_3, k_1, k_2, T)$ streaming code
- Let $n = \max(n_1, n_2, n_3)$. Rate pair $(R_1, R_2) = (\frac{k_1}{n}, \frac{k_2}{n})$

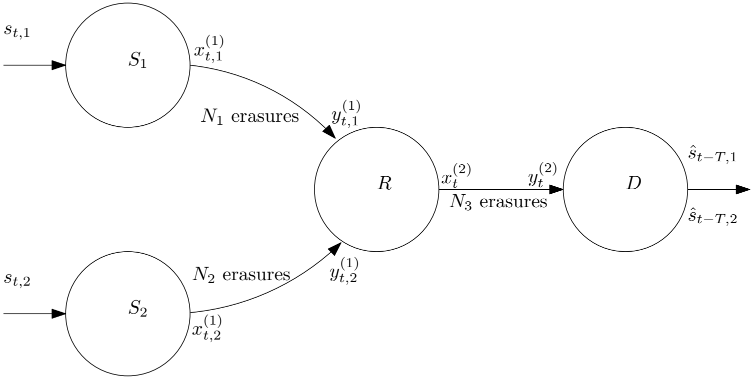


Figure: Multiaccess relay network (Facenda et al., 2023a)

Rate Region

- Assume WLOG $N_1 \geq N_2$
- Single user point to point bounds still apply:

$$R_1 \leq C(T - N_3, N_1)$$
$$R_2 \leq C(T - N_3, N_2)$$

- Additional bound on the sumrate $R_1 + R_2$:

$$R_1 + R_2 \leq C(T - N_2, N_3)$$

Nonadaptive Scheme 1 (CSWDF)

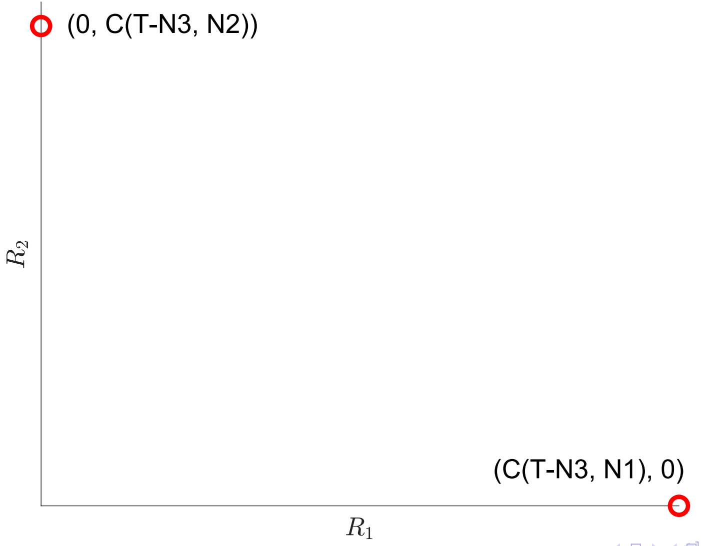
- Concatenated Symbol-wise Decode-and-Forward (CSWDF) concatenates two single user SWDF codes
- Main Idea: Timesharing

Lemma (Facenda et al., 2023a)

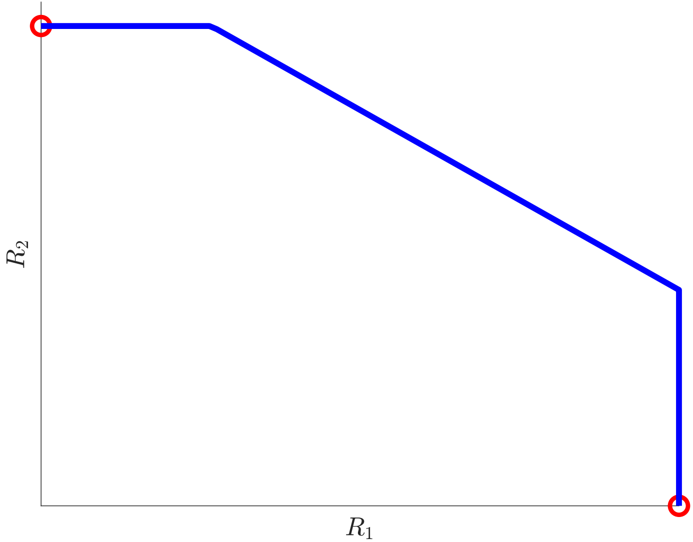
Given two rate pairs (R_1, R_2), (R_1', R_2'), we can construct codes for any convex combination through concatenation.

- Construct codes for $(C(T - N_3, N_1), 0)$ and $(0, C(T - N_3, N_2))$ and timeshare to obtain the full rate region

CSWDF



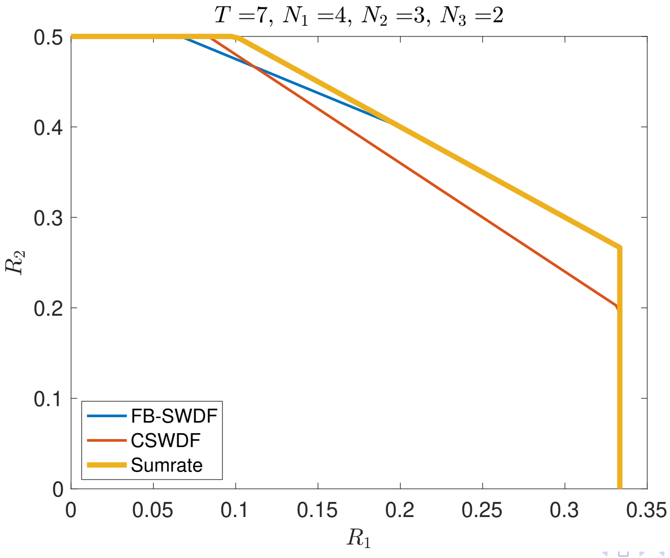
CSWDF



Nonadaptive Scheme 2 (FB-SWDF)

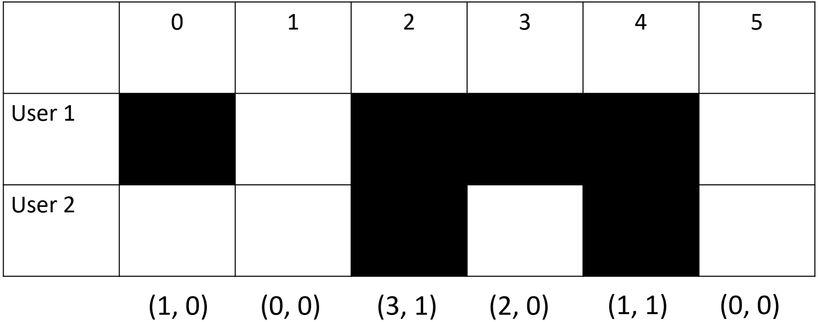
- Fixed Bottleneck Symbol-Wise Decode-and-Forward (FB-SWDF) encodes source symbols jointly
- Joint encoding better since specific to multiaccess setting
- Can achieve higher rates than CSWDF

Nonadaptive schemes rate region



Adaptive Multiaccess Codes

- Considering both users, larger number of possible erasure patterns
- Associate to each time an erasure pattern (N_1', N_2'), $0 \leq N_1' \leq N_1, 0 \leq N_2' \leq N_2$.



Adaptive Scheme 1 (Concatenated Adaptive)

- Using similar idea as CSWDF, concatenate two single user adaptive codes
- Use subset adaptation at the two endpoints and timeshare between the two points
- Using subset scheme instead of adaptive allows the code to be practical
- Baseline adaptive scheme

Adaptive Scheme 2 (Joint Adaptive)

- Use a subset based approach but over both users: symbols from users encoded jointly
- Define j as the rate of adaptation if all erasure patterns (N_1', N_2') with $\max(N_1', N_2') \leq j$ can be adapted to
- To adapt, relay groups symbols together and sends symbols from time i at time $i + j$ or $i + N_2$
- Source to Relay codes can be as in CSWDF or FB-SWDF
- Joint adaptation can allow for dealing with additional erasure patterns concatenated cannot e.g. $(0, s_2)$.

Joint vs Concatenated (Example)

- Consider the following S-R code and response of joint and concatenated

Table: S-R Codes, $T = 4, N_1 = 3, N_2 = 2, N_3 = 1, j = 0$

	Time i	1	2	3	4	5	6
		$a_1[1:4]$	a_2	a_3	a_4	a_5	a_6
User 1			$p_{1,1}$	$p_{2,1}$			
				$p_{1,2}$	$p_{2,2}$		
					$p_{1,3}$	$p_{2,3}$	
User 2		$c_1[1:4]$	c_2	c_3	c_4	c_5	c_6
		$d_1[1:4]$	d_2	d_3	d_4	d_5	d_6
			$p_{1,1}$	$p_{2,1}$			
				$p_{1,2}$	$p_{2,2}$		

Table: Adaptive R-D Code (Concat)

Time i	1	2	3	4	5	6	
				$a_1[1:4]$			4
					$p_1[1:4]$		4
	$c_1[1:2]$						2
		$c_1[3:4]$					2
			$d_1[1:2]$				2
				$d_1[3:4]$			2
					$q_1[1:2]$		2

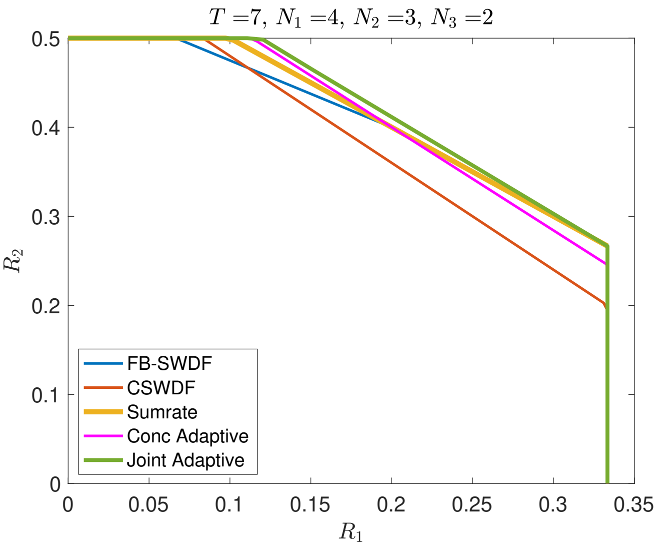
Table: Adaptive R-D Code (Joint)

Time i	1	2	3	4	5	6	
	$c_1[1:3]$						3
		$c_1[4] \frown d_1[1:2]$					3
			$d_1[3:4] \frown a_1[1]$				3
				$a_1[2:4]$			3
					$q_1[1:3]$		3

Joint Adaptive

- Choose rate pair achieved by CSWDF: $(R_1, R_2) = (\frac{Ak_1}{n}, \frac{Bk_2'}{n})$
- If $\max(N_1', N_2') \leq j$, relay sends $\frac{Ak_1+Bk_2'}{T+1-N_3-j}$ per timeslot
- If number of available symbols larger than $\frac{Ak_1+Bk_2'}{T+1-N_3-j}$, also use the same rate
- Otherwise, send $\frac{Ak_1}{T+1-N_3-N_1} + \frac{Bk_2'}{T+1-N_3-N_2}$
- Adapting decreases value of n_3. Then with $n = \max(n_1, n_2, n_3)$, can decrease n and increase R_1, R_2

Results



T =7, N_1 =4, N_2 =3, N_3 =2
0.5
0.4
0.3
0.2
0.1
0
R_2
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
R_1
FB-SWDF
CSWDF
Sumrate
Conc Adaptive
Joint Adaptive

Next Steps

- Other options for adaptive schemes
- Best choice of codes to use for S-R links
- More analysis on the performance of the schemes in different cases
- Characterizing gains achieved by joint adaptation over concatenated

Bibliography I

Facenda, Gustavo Kasper, Elad Domanovitz, Ashish Khisti, Wai-Tian Tan, and John Apostolopoulos (2023a). "Streaming Erasure Codes Over Multi-Access Relayed Networks". In: *IEEE Transactions on Information Theory* 69.2, pp. 860–885. DOI: 10.1109/TIT.2022.3214165.

Facenda, Gustavo Kasper, M. Nikhil Krishnan, Elad Domanovitz, Silas L. Fong, Ashish Khisti, Wai-Tian Tan, and John Apostolopoulos (2023b). "Adaptive relaying for streaming erasure codes in a three node relay network". In: *IEEE Transactions on Information Theory*, pp. 1–1. DOI: 10.1109/TIT.2023.3254464.

Fong, Silas L., Ashish Khisti, Baochun Li, Wai-Tian Tan, Xiaoqing Zhu, and John Apostolopoulos (2019). "Optimal Streaming Erasure Codes over the Three-Node Relay Network". In: *2019 IEEE International Symposium on Information Theory (ISIT)*, pp. 3077–3081. DOI: 10.1109/ISIT.2019.8849846.

Upper Bound (SWDF)

- Main idea: Consider three-node network as two point-to-point links with reduced delay
- Delay for first link is $T - N_2$ since it must handle a burst of N_2 in second link:

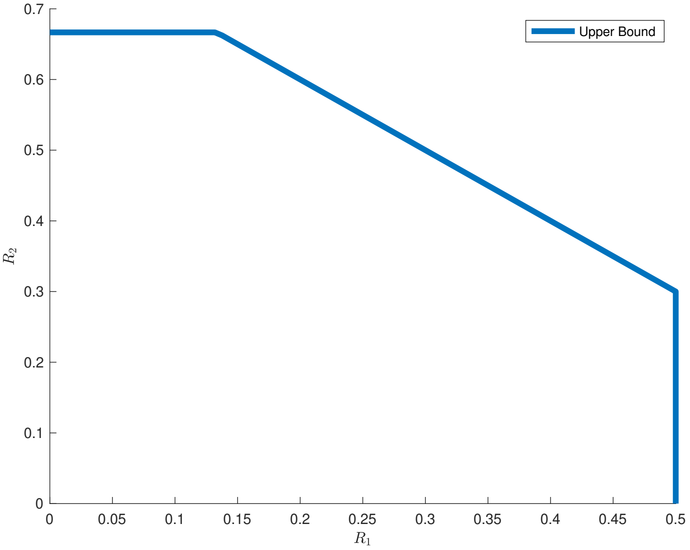
$$C(T, N_1, N_2) \leq C(T - N_2, N_1) = \frac{T + 1 - N_2 - N_1}{T + 1 - N_2} \tag{1}$$

- Similarly,

$$C(T, N_1, N_2) \leq C(T - N_1, N_2) = \frac{T + 1 - N_1 - N_2}{T + 1 - N_1} \tag{2}$$

- $C(T, N_1, N_2) = \min\{C(T - N_2, N_1), C(T - N_1, N_2)\}$
- SWDF optimal as a *nonadaptive* scheme

Rate Region (plot)



FB-SWDF Details

- Fixed Bottleneck Symbol-Wise Decode-and-Forward (FB-SWDF) fixes the code for relay and user 1 to have maximal rate
- Constructs user 2 code to match delay constraints
- Goal is to achieve the sumrate capacity

FB-SWDF Example

- Consider $T = 4, N_1 = 3, N_2 = 2, N_3 = 1$
- Fix R-D code to have rate $C(T - N_2, N_3)$: Use $(3, 2)$ code concatenated 4 times
- Fix S1-R code to have rate $C(T - N_3, N_1)$: Use $(4, 1)$ concatenated 3 times
- S2-R then has a constraint on how many symbols it can send per delay

FB-SWDF Example (cont'd)

Table: User 1, 2 codes

Time	0	1	2	3	4	5	6
$a_i[1:3]$	a_0	a_1	a_2	a_3	a_4	a_5	a_6
$a_{i-1}[1:3]$		a_0	a_1	a_2	a_3	a_4	a_5
$a_{i-2}[1:3]$			a_0	a_1	a_2	a_3	a_4
$a_{i-3}[1:3]$				a_0	a_1	a_2	a_3

Time	0	1	2	3	4	5	6
$b_i[1:4]$	b_0	b_1	b_2	b_3	b_4		
$b_{i-1}[1:4]$		b_0	b_1	b_2	b_3		
$b_{i-2}[1:4]$			b_0	b_1	b_2	b_3	
$c_{i-1}[1]$		c_0	c_1	c_2	c_3		
$c_{i-2}[1]$			c_0	c_1	c_2	c_3	
$c_{i-3}[1]$				c_0	c_1	c_2	c_3

Table: Relay to Destination code

Time	2	3	4	5	6
$b_i[1:4]$	b_0	b_1	b_2	b_3	b_4
$a_{i-1}[1:3] \frown c_{i-1}[1]$		$a_0 \frown c_0$	$a_1 \frown c_1$	$a_2 \frown c_2$	$a_3 \frown c_3$
$b_i + a_{i-1} \frown c_{i-1}$			$b_0 + a_0 \frown c_0$	$b_1 + a_1 \frown c_1$	$b_2 + a_2 \frown c_2$

Joint Adaptive v1

- Choose two corner points from FB-SWDF scheme
- At each, nonadaptive construction gives value of k_1, k_2
- At time i, consider erasure pattern (N_1', N_2')
- If $\max(N_1', N_2') \leq j$, relay sends $\frac{k_1+k_2'}{T+1-N_3-j}$ per timeslot
- Otherwise, send $\frac{k_1+k_2}{T+1-N_3-N_2}$
- Source to relay codes remain unchanged, relay to destination modified
- Calculate new value of n_3.
- Timeshare between corner points and adaptive values of n_3

Simulations