Codes for the Two–user Binary Adder Channel

Daniele Fabio Fuccio

danfu@cc.hut.fi

Helsinki University of Technology
Institute of Digital Communications – Communications Laboratory
P.O. Box 3000, FIN-02015 HUT – Finland

Supervisor: Prof. Patric R. J. Östergård

Outline

✓ The binary adder channel (BAC)

Outline

- ✓ The binary adder channel (BAC)
- ✓ Capacity region and some bounds

Outline

- ✓ The binary adder channel (BAC)
- ✓ Capacity region and some bounds
- ✓ Uniquely decodable codes (UD) and equivalent codes

Outline

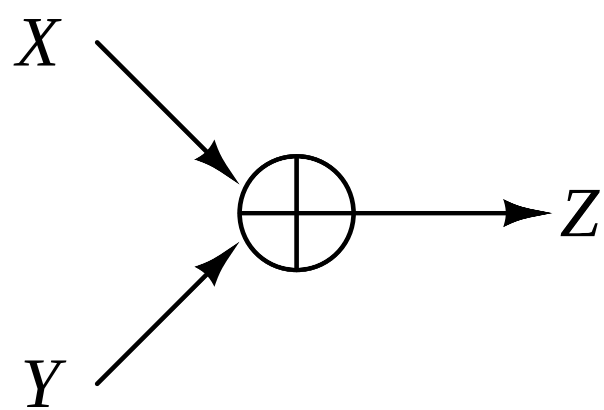
- ✓ The binary adder channel (BAC)
- ✓ Capacity region and some bounds
- ✓ Uniquely decodable codes (UD) and equivalent codes
- ✓ Method to approach the results

Outline

- ✓ The binary adder channel (BAC)
- ✓ Capacity region and some bounds
- ✓ Uniquely decodable codes (UD) and equivalent codes
- ✓ Method to approach the results
- ✓ Results and Conclusions

The BAC

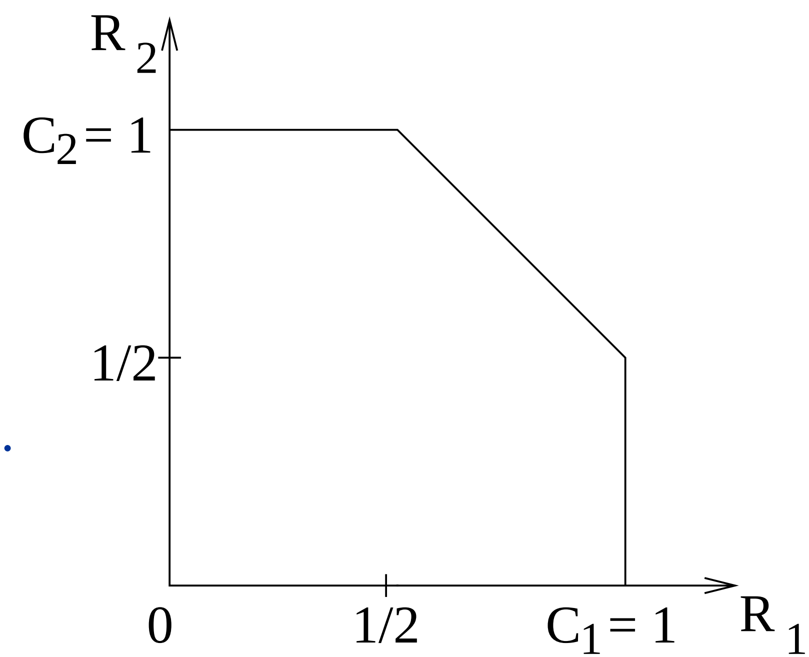
- ✓ This channel has binary inputs, $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and a ternary output $\mathcal{Z} = \{0, 1, 2\}$. There is no ambiguity in (X, Y) if $Z = 0$ or $Z = 2$ is received; but $Z = 1$ can result from either (0,1) or (1,0)



Shannon capacity region (C_S)

✓ The **Shannon capacity region** is obtained under the assumption to use codes with length that tends to infinity

$$0 \le R_1 \le 1,$$
$$0 \le R_2 \le 1,$$
$$R_1 + R_2 \le 3/2.$$



UD and inequivalent codes

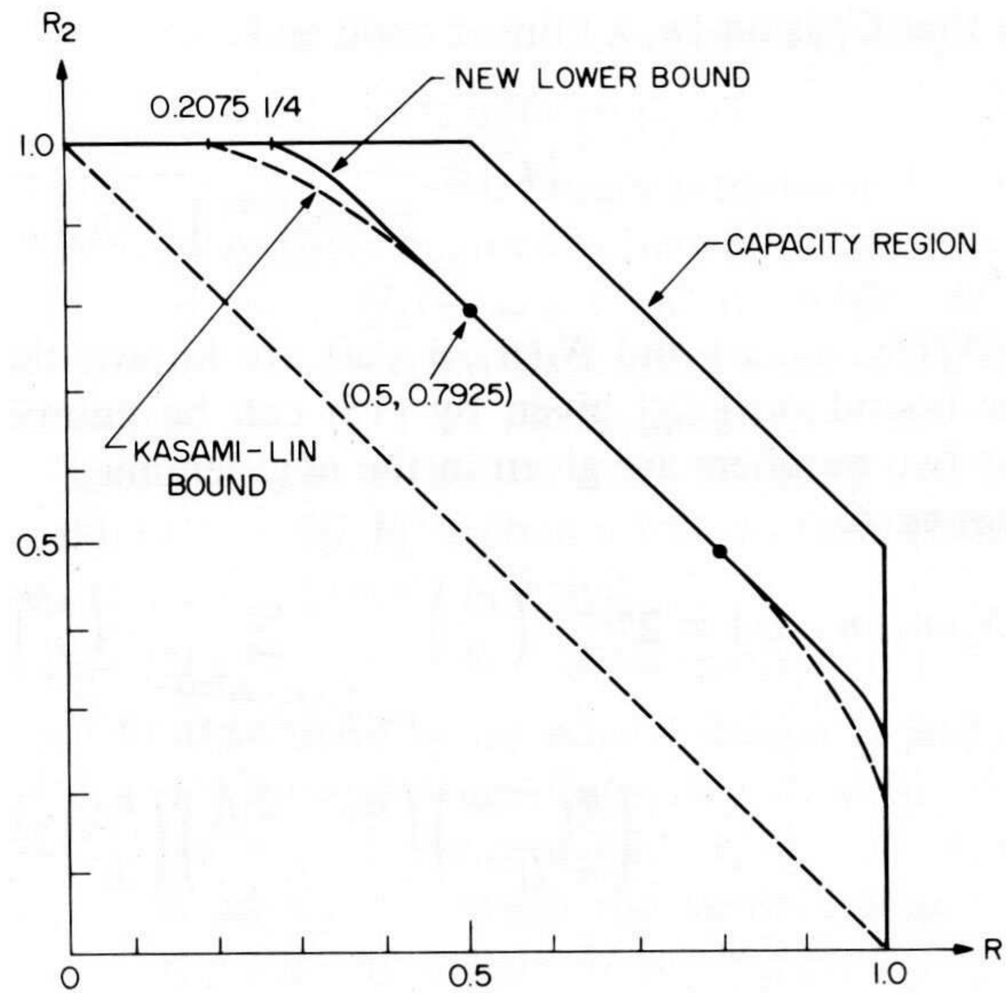
- ✓ The code pair $(\mathcal{C}_1, \mathcal{C}_2)$ is called **uniquely decodable** (UD) if the sums $c_1 + c_2$ of all pairs $(c_1, c_2) \in (\mathcal{C}_1 \times \mathcal{C}_2)$ are all different

- ✓ Two codes are said to be **equivalent** if there is a permutation of the coordinates (bits of the codeword) together with **n** permutations of the coordinate values, one for each of the coordinates that map one code into the other

Zero–error capacity region (C_S)

- ✓ In communication systems it is important to require to transmit over a noiseless synchronous 2–user BAC with error probability **strictly** zero
- ✓ The **zero–error capacity region** (C_{ZE}), it is the convex closure of all rate pairs (R_X, R_Y) which corresponds to uniquely decodable codes

KLWY lower bound

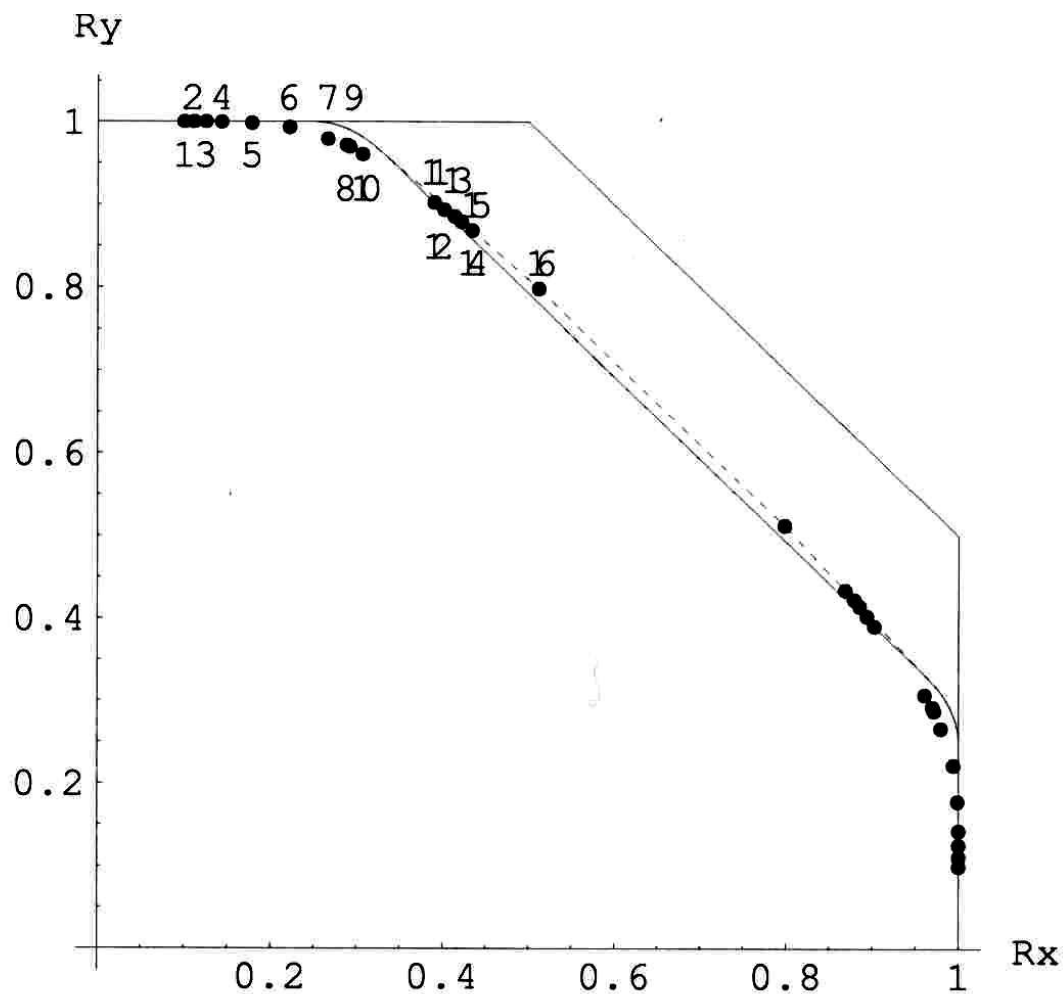


CT–construction results

nr.	N	R_1	R_2	$R_1 + R_2$
a1	100	0.10000	0.99999	1.09999
a2	81	0.11111	0.99998	1.11109
a3	64	0.12500	0.99992	1.12492
a4	49	0.14286	0.99968	1.14254
b1	342	0.17823	0.99844	1.17667
a7	115	0.18622	0.99749	1.18370
b3	174	0.26662	0.97899	1.24561
b4	320	0.28796	0.97166	1.25962

nr.	N	R_1	R_2	$R_1 + R_2$
b5	350	0.29185	0.96981	1.26165
b6	848	0.30695	0.96074	1.26769
b7	240	0.40224	0.89375	1.29599
b8	448	0.41425	0.88527	1.29951
b9	512	0.42204	0.87913	1.30116
b10	800	0.43425	0.86873	1.30298
a22	48	0.44514	0.85820	1.30334
b11	1344	0.44310	0.86056	1.30366

Non–constructive lower bound



The best known UD code

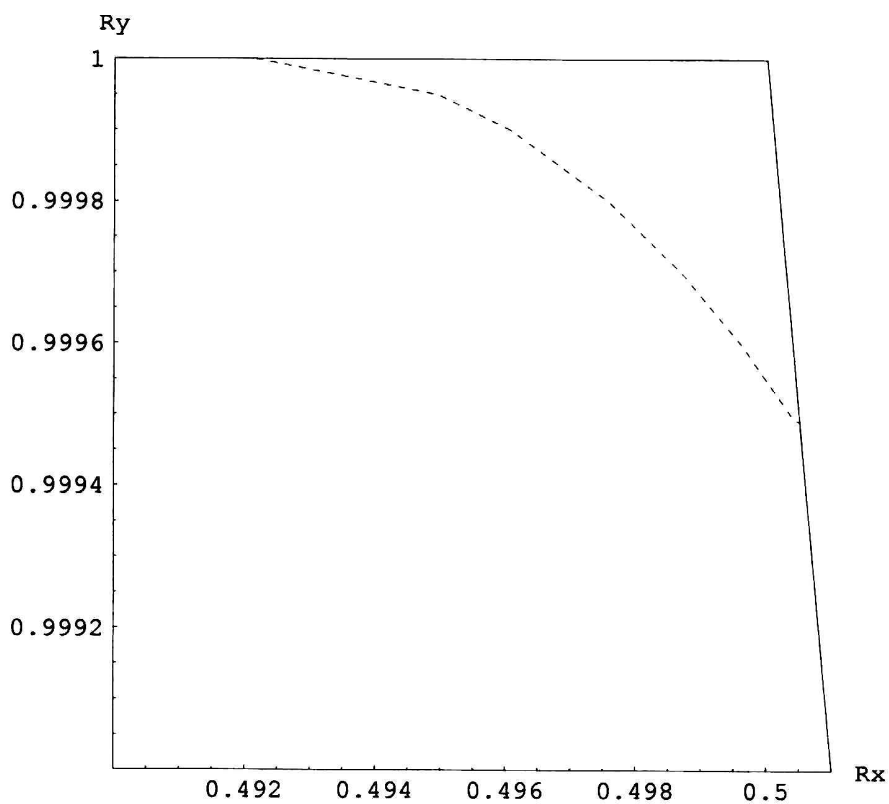
- ✓ The best known UD code pair is obtained from a code of length 7, the sizes are $|\mathcal{C}_1| = 12$ and $|\mathcal{C}_2| = 48$

$$\mathcal{C}_1 = \{1, 13, 19, 34, 40, 52, 75, 87, 93, 108, 114, 126\},$$
$$\mathcal{C}_2 = \{6, 10, 14, 15, 16, 20, 21, 24, 26, 27, 30, 38, 39, 43, 46, 47, 49, 52, 53, 56, 57, 58, 59, 62, 63, 64, 65, 66, 68, 70, 71, 74, 80, 88, 89, 96, 97, 99, 100, 101, 102, 103, 106, 107, 112, 113, 120, 121\}$$

$$R_1 + R_2 = 1.30999 \text{ bits per transmission.}$$

Urbanke–Li's upper bound

- ✓ The zero–error capacity region is **strictly** smaller than the Shannon capacity region.



Maximum clique problem

- ✓ **Clique** of a graph G
- ✓ The size of a largest clique of G is called the **clique number** of G
- ✓ Maximum clique problem is NP–hard

An improved result

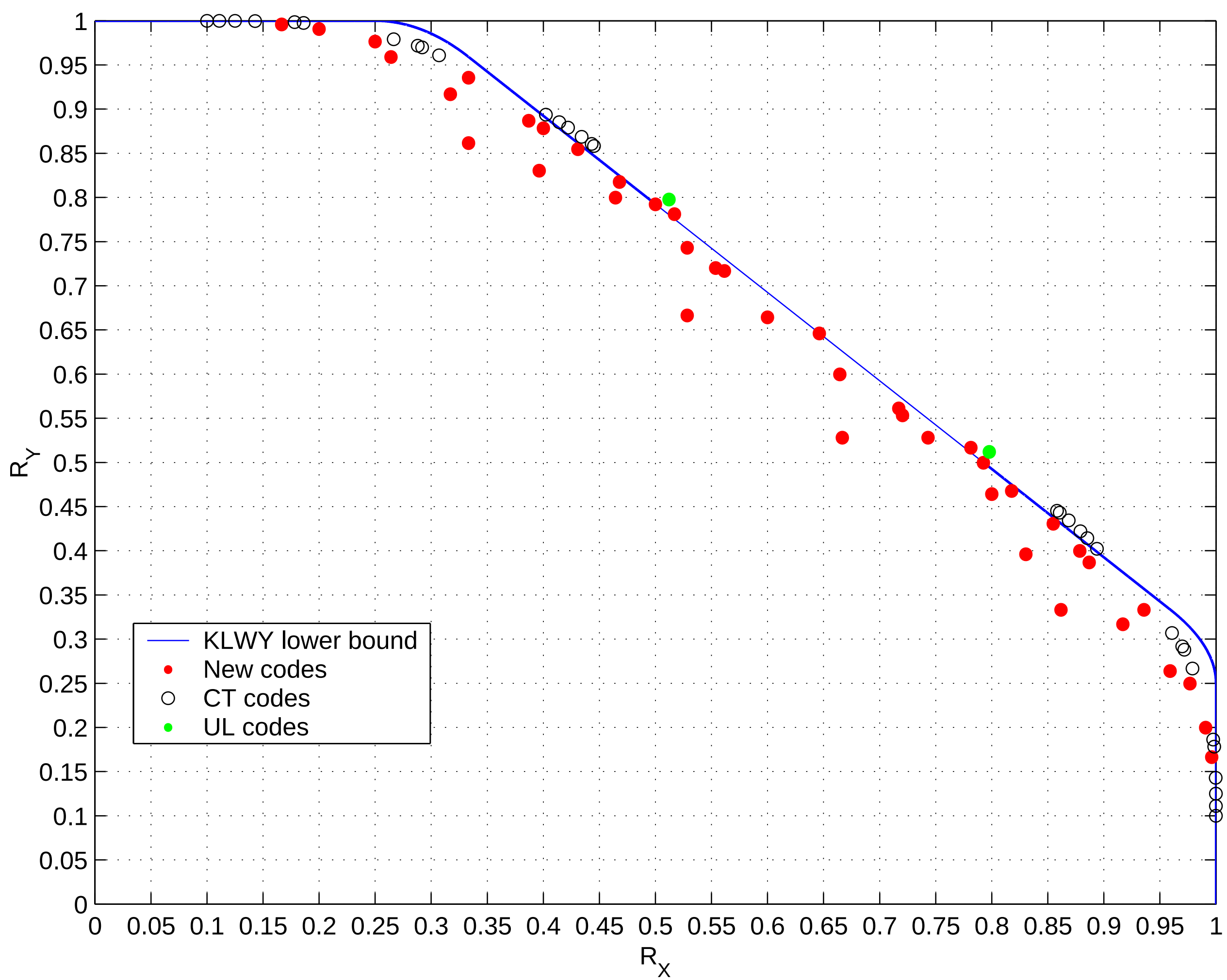
- ✓ Coebergh found a code pair $(\mathcal{C}_1, \mathcal{C}_2)$ of length 7 and sizes 12 and 47 with $R_1 + R_2 = 1.30565$ bits per transmission

$$\mathcal{C}_1 = \{1, 4, 10, 19, 28, 31, 96, 99, 108, 117, 123, 126\}$$

- ✓ $\mathcal{C}_2$ has been improved:

$$\mathcal{C}_2 = \{6, 7, 9, 10, 13, 15, 16, 18, 21, 22, 24, 25, 38, 39, 41, 42, 43, 45, 47, 48, 50, 53, 54, 55, 56, 57, 58, 70, 71, 73, 74, 75, 77, 79, 80, 82, 85, 86, 87, 88, 89, 90, 103, 106, 109, 112, 118, 121\}$$

Best obtained results



Best obtained results

n	$\lvert\mathcal{C}_1\rvert$	$\mathcal{C}_1$	Sum–rate
2	2	0 3	1.292481
2	3	0 1 2	1.292481
3	2	0 7	1.269118
3	7	0 1 2 3 4 5 6	1.269118

n	$\lvert\mathcal{C}_1\rvert$	$\mathcal{C}_1$	Sum–rate
4	4	0 3 12 15	1.292481
4	6	0 1 2 7 13 14	1.292481
4	6	0 1 2 12 13 14	1.292481
4	6	0 1 2 13 14 15	1.292481
4	6	0 1 6 10 13 15	1.292481
4	9	0 1 2 3 4 5 8 10 12	1.292481
4	9	0 1 2 3 4 5 10 12 14	1.292481

Best obtained results

n	$\lvert\mathcal{C}_1\rvert$	$\mathcal{C}_1$	Sum–rate
4	9	0 1 2 3 4 11 12 13 15	1.292481
4	9	0 1 2 4 9 10 11 12 13	1.292481
4	9	0 1 2 4 9 10 13 14 15	1.292481
4	9	0 1 2 5 6 11 12 13 14	1.292481
5	6	0 3 12 21 26 31	1.298371
5	6	0 3 12 21 27 30	1.298371
5	15	0 1 2 3 4 7 8 15 16 23 24 28 29 30 31	1.298371
5	15	0 1 2 3 4 8 12 19 21 22 23 27 28 29 30	1.298371
5	15	0 1 2 3 4 9 12 13 18 22 26 27 28 29 30	1.298371

Conclusions

✓ Good results compared to the solved code length

Conclusions

- ✓ Good results compared to the solved code length
- ✓ Exhaustive research has been completed for code length $n = 2, 3, 4, 5$ and code length 6 only partially (computational reasons)

Conclusions

- ✓ Good results compared to the solved code length
- ✓ Exhaustive research has been completed for code length $n = 2, 3, 4, 5$ and code length 6 only partially (computational reasons)
- ✓ Multiuser coding can achieve **higher total rate** of transmission (sum–rate) than traditional channel multiplexing techniques such as time–division

Conclusions

- ✓ Good results compared to the solved code length
- ✓ Exhaustive research has been completed for code length $n = 2, 3, 4, 5$ and code length 6 only partially (computational reasons)
- ✓ Multiuser coding can achieve **higher total rate** of transmission (sum–rate) than traditional channel multiplexing techniques such as time–division
- ✓ This work could find application in a T–user BAC