

Efficient Encoding of Updated Information

**Jack Keil Wolf
UCSD, La Jolla, CA
(P.I.B. 1965-1973)**

**WICAT Workshop on
Cooperative Communications**

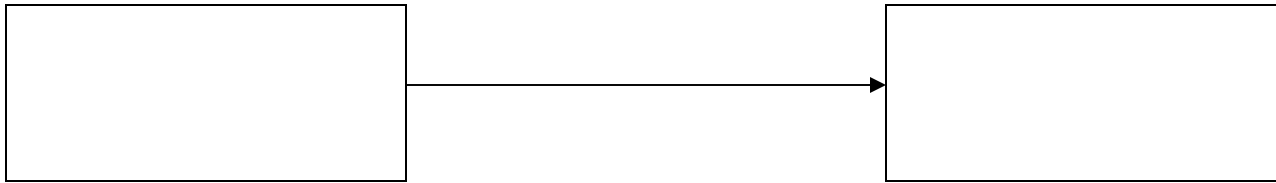
October 21, 2005

PROBLEM STATEMENT

- Information at one terminal (called the transmitter) is to be conveyed **efficiently** to one or more other terminals (called the receivers).
- The receivers have **closely related versions** of the information to be transmitted.
- This is an example of source encoding with side information at the receivers.

OUR PROBLEM (Two terminal Version)

We begin with one transmitter and one receiver.



Transmitting terminal

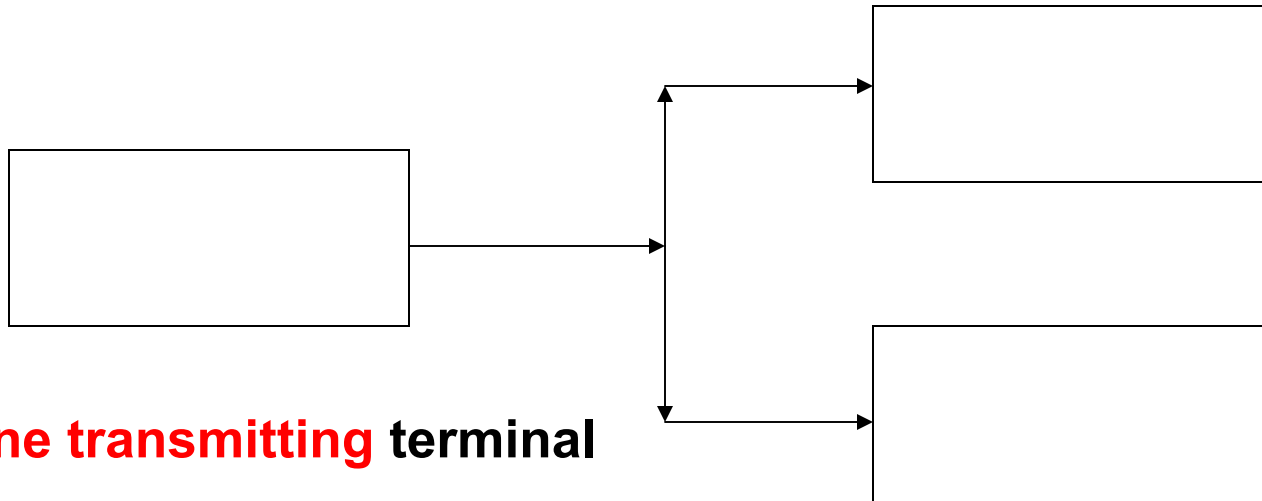
Contains information
to be transmitted.

Receiving terminal

Contains related
information.

BROADCAST VERION OF OUR PROBLEM

- We then generalize to the problem of one transmitter and two or more receivers.



One transmitting terminal

Contains information to be transmitted.

Two or more receiving terminals

Each contains information closely related to the transmitter's Information.

TOY EXAMPLE

- The talk uses a simple **toy** example to illustrate the basic ideas.
- In this toy example we first assume there is only one receiving terminal.
- Then we generalize to the case where there are two or more receiving terminals.

TOY EXAMPLE

(1 RECEIVING terminal)

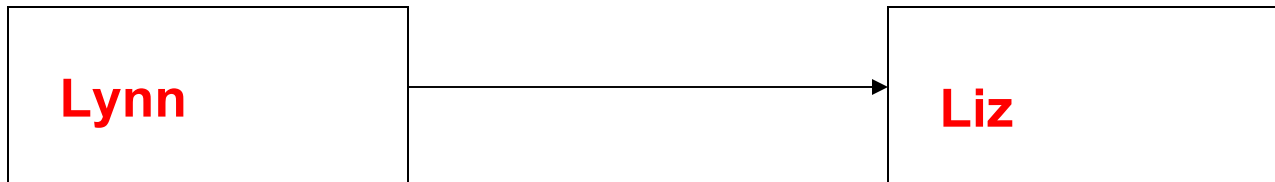
- In Brooklyn, NY and Elizabeth, NJ, we assume that the weekly weather is stored in terminals as binary 7-vectors where a “1” means “good weather” and a “0” means “bad weather”.
- The weather in Brooklyn is stored in a terminal named **Lynn** as $\underline{X} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$.
- The weather in Elizabeth, N.J. is stored in a terminal named **Liz** as $\underline{Y} = (y_1, y_2, y_3, y_4, y_5, y_6, y_7)$.

TOY EXAMPLE (1 RECEIVING terminal)

- We assume that the daily weather in Brooklyn is i.i.d. and equally likely to be 0 or 1.
- We assume the same is true for the weather in Elizabeth.
- However, we assume that the weather in Brooklyn and Elizabeth is so similar that for each and every week, X and Y differ in at most one position.

TOY EXAMPLE

(1 RECEIVING terminal)



Brooklyn

Weather X

Elizabeth

Weather Y

TOY EXAMPLE

(1 RECEIVING terminal)

- Lynn is to transmit information about X to Liz.
- It would seem that this would require the transmission of 7 binary digits.
- However, if Lynn knows both X and Y, it is obvious that she can transmit the information in only 3 binary digits.

TOY EXAMPLE

(1 RECEIVING terminal)

- If Lynn knows both X and Y, she can transmit the difference (if any) between X and Y. That is, she sends:
 - (000) if X=Y
 - (001) if X differs from Y in the first position,
 - (010) if X differs from Y in the second position,
 - ...
 - (111) if X differs from Y in the 7th position.
- Liz can compute X from this difference since she already knows Y.

TOY EXAMPLE

(1 RECEIVING terminal)



Transmitter in Brooklyn

Receiver in Elizabeth

Knows **both** X and Y

Initially knows Y

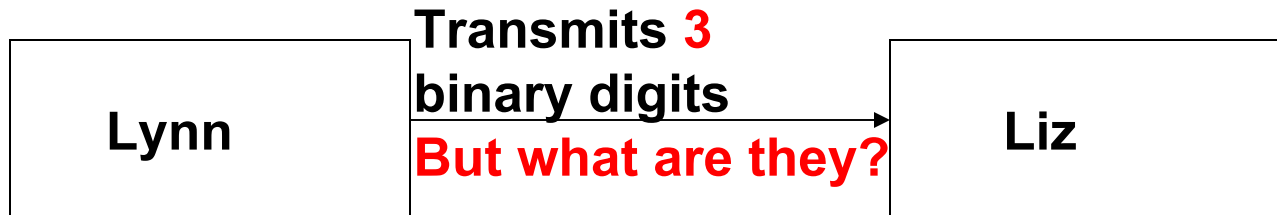
Learns X after receipt
of **only 3** binary digits.

TOY EXAMPLE

(1 RECEIVING terminal)

- But what if Lynn knows X but **does not know** Y?
- Lynn then does not know how X and Y relate to each other.
- In this case, it seems impossible for Lynn to transmit just 3 binary digits to Liz so that Liz would learn X.
- But as will be shown later, the **impossible is possible!!!**

TOY EXAMPLE (1 RECEIVING terminal)



Transmitter in Brooklyn

Receiver in Elizabeth

Knows **only** X

Initially knows Y
But learns about X.

TOY EXAMPLE

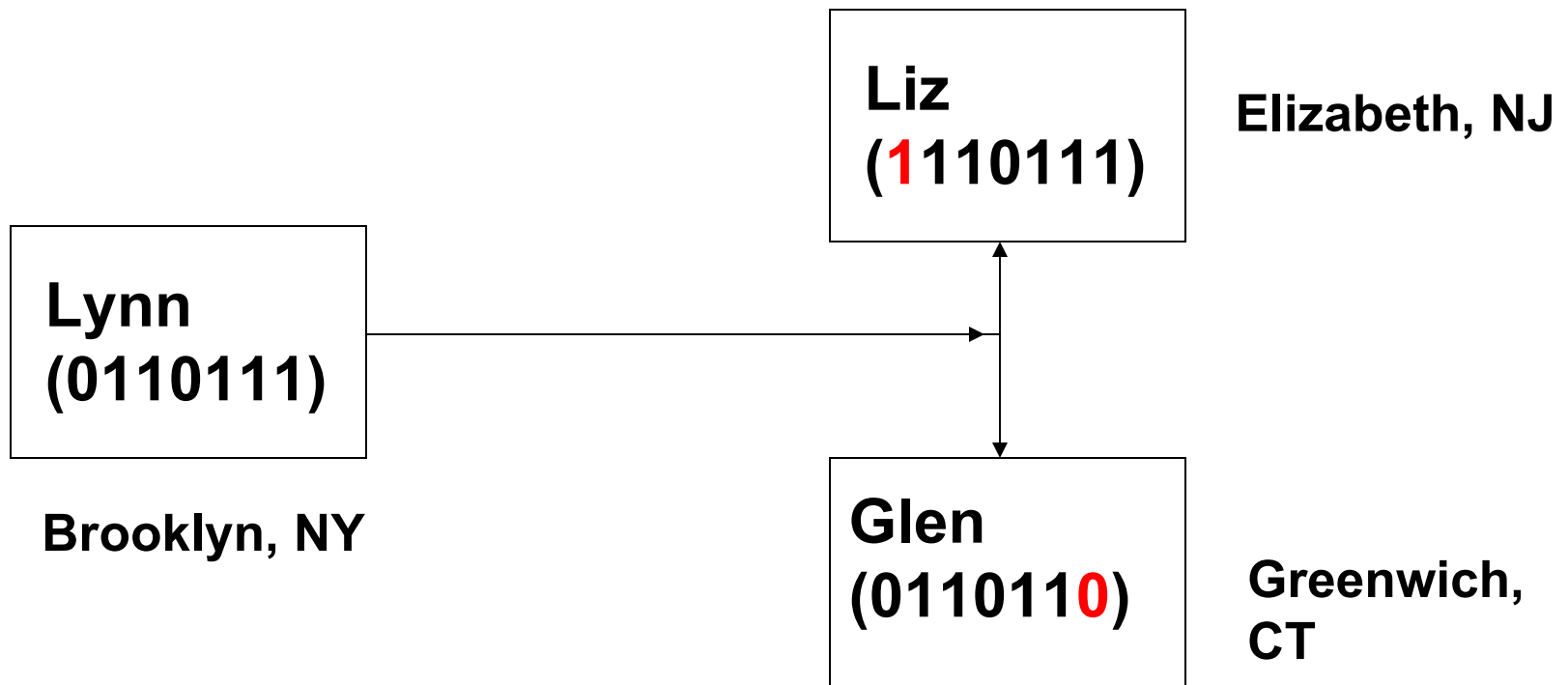
(2 RECEIVING terminals)

- But first, we extend this toy example to the case of 3 cities: Brooklyn, NY, Elizabeth, NJ and Greenwich, CT (where the terminals are now named **Lynn**, **Liz** and **Glen**, respectively).
- The weather in these 3 cities are denoted by the vectors, \underline{X} , \underline{Y} , and \underline{Z} , respectively.
- We assume that \underline{X} and \underline{Y} differ in at most one position and also that \underline{X} and \underline{Z} differ in at most one position.
- Note, however that we are **not** assuming that \underline{Y} and \underline{Z} are equal. As a matter of fact, \underline{Y} and \underline{Z} can differ in **as many as 2** positions.

TOY EXAMPLE

(2 RECEIVING terminals)

- The problem is for Lynn to **broadcast** information to Liz and Glen such that **both** Liz and Glen can learn X.



TOY EXAMPLE

(2 RECEIVING terminals)

- If Lynn knows X, Y and Z, she can broadcast the difference (if any) between X and Y and also the difference (if any) between X and Z.
- This would seem to require that $3+3=6$ binary digits be broadcast.
- But we will soon see that she can do better than this. Specifically, **only 3** binary digits need be broadcast.

TOY EXAMPLE: (1 RECEIVING terminal)

- We return to the problem of just one transmitting terminal and one receiving terminal, Lynn and Liz.
- As before we assume that Liz knows Y but now we assume that Lynn does **not** know Y.?
- Lynn of course knows her own weather, X.
- We next demonstrate a scheme where even though Lynn does **not** know Y, she can still send only 3 binary digits to Liz and allow Liz to learn X.

TOY EXAMPLE

(1 RECEIVING terminal)

- Lynn creates a **hash function** that compresses the 7 binary digits, \underline{X} , into 3 binary digits $(s_1, s_2, s_3) = \underline{S}$, where the components of the vector \underline{S} are given by the equations:

$$\begin{aligned} S_1 &= X_4 \oplus X_5 \oplus X_6 \oplus X_7 \\ S_2 &= X_2 \oplus X_3 \oplus X_6 \oplus X_7 \\ S_3 &= X_1 \oplus X_3 \oplus X_5 \oplus X_7 . \end{aligned}$$

Here the “ \oplus ” sign indicates modulo 2 addition.

- Lynn transmits **the 3 binary digits**, \underline{S} , to Liz who already knows \underline{Y} .

TOY EXAMPLE

(1 RECEIVING terminal)

- Liz uses the same hash function on \underline{Y} to create the vector $\underline{T}=(t_1,t_2,t_3)$ where the components of \underline{T} are given by the equations:

$$\begin{aligned}t_1 &= y_4 \oplus y_5 \oplus y_6 \oplus y_7 \\t_2 &= y_2 \oplus y_3 \oplus y_6 \oplus y_7 \\t_3 &= y_1 \oplus y_3 \oplus y_5 \oplus y_7 .\end{aligned}$$

- Finally Liz adds the hash functions \underline{S} and \underline{T} (bit by bit, modulo 2) to obtain the composite hash function $\underline{W}=(\underline{S} \oplus \underline{T}) = (w_1,w_2,w_3)$.

TOY EXAMPLE

(1 RECEIVING terminal)

- From W, Liz then can find which component of Y to complement to find X.
- The decoding rule used by Liz is:
 - if W=(000), then X=Y,
 - if W=(001), the 1st bit should be complemented,
 - if W=(010), the 2nd bit should be complemented,
 - ...
 - if W=(111), the 7th bit should be complemented.

TOY EXAMPLE

(1 RECEIVING terminal)

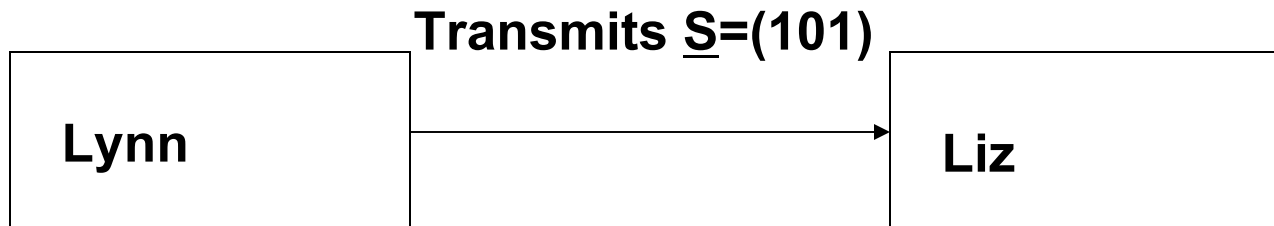
- Numerical example: Let $\underline{X}=(1110100)$, and $\underline{Y}=(11101\mathbf{1}0)$. Note that \underline{X} and \underline{Y} differ in the 6th bit.

$$\begin{array}{rcccccccc}
 s_1 & = & x_4 & \oplus & x_5 & \oplus & x_6 & \oplus & x_7 \\
 s_2 & = & x_2 & \oplus & x_3 & \oplus & x_6 & \oplus & x_7 \\
 s_3 & = & x_1 & \oplus & x_3 & \oplus & x_5 & \oplus & x_7 .
 \end{array}$$

- Lynn computes $\underline{S}=(101)$ and sends it to Liz. Liz computes $\underline{T}=(0\bar{1}\bar{1})$ and then $\underline{W}=(\underline{S} \oplus \underline{T})=(110)$.
- Since $W=(110)$, Liz knows that the 6th bit of \underline{Y} should be complemented in order to obtain \underline{X} .

TOY EXAMPLE

(1 RECEIVING terminal)



$$\underline{X}=(1110100)$$

$$\text{Computes: } \underline{S}=(101)$$

$$\underline{Y}=(1110110)$$

$$\text{Computes: } \underline{T}=(011)$$

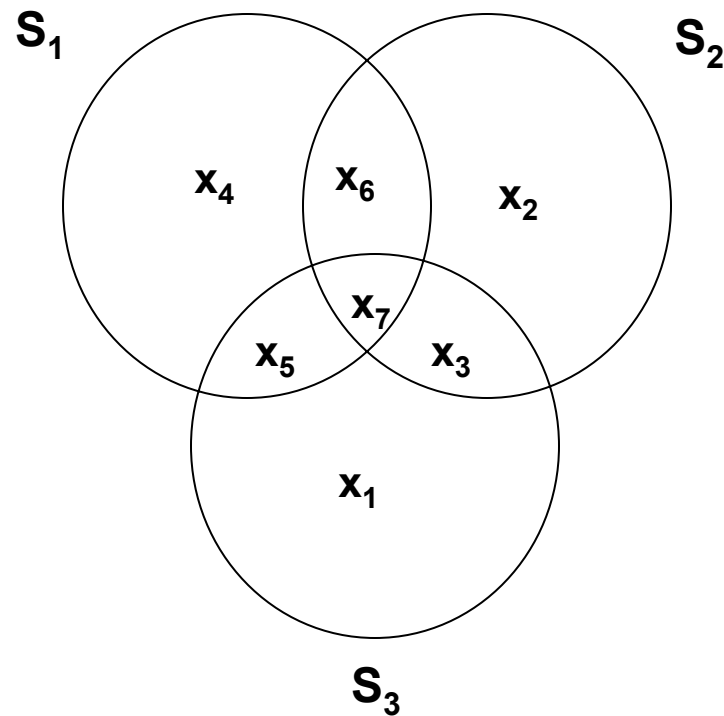
$$\text{Then: } \underline{W}=(101) \oplus (011) = (110)$$

$$\text{Then: } \underline{E}=(0000010)$$

$$\begin{aligned} \underline{X} &= (1110110) \oplus (0000010) \\ &= (1110100) \end{aligned}$$

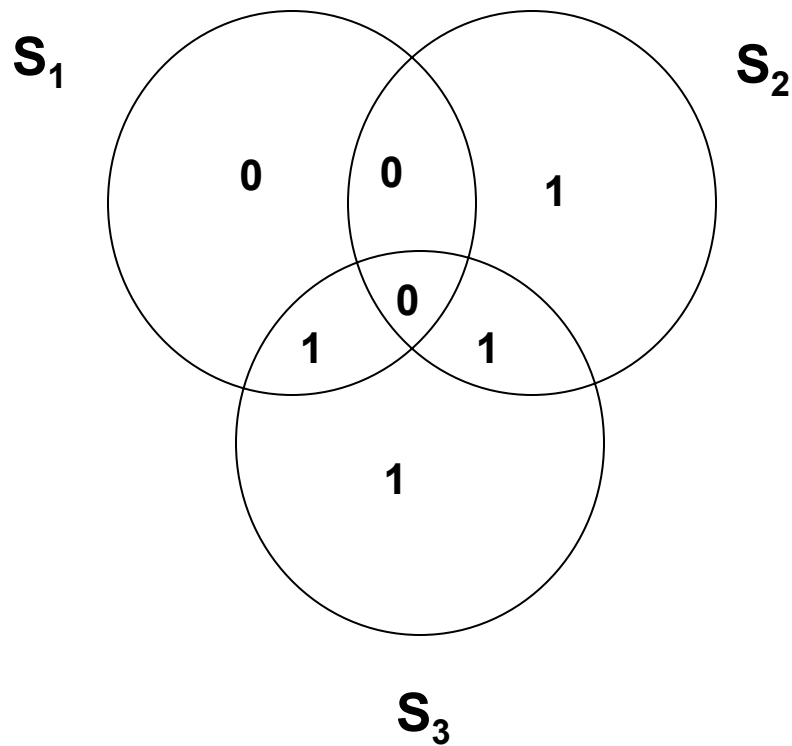
EXPLANATION OF TOY EXAMPLE a la McEliece

X=(1110100)

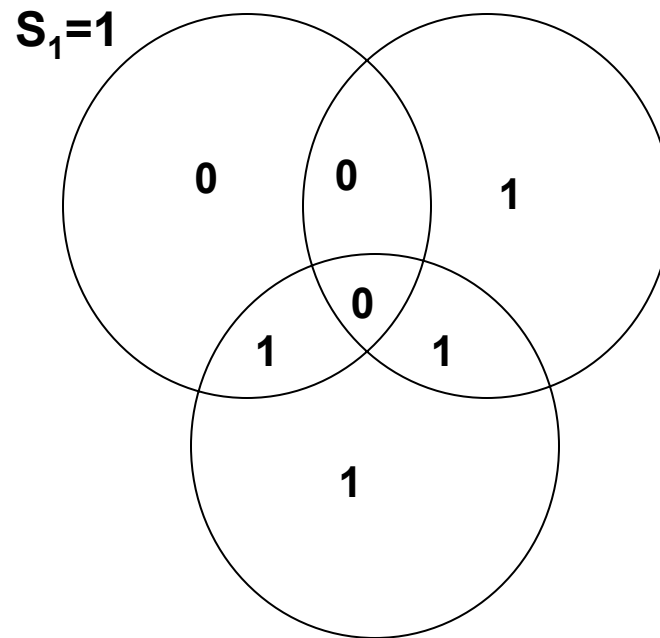


EXPLANATION OF TOY EXAMPLE a la McEliece

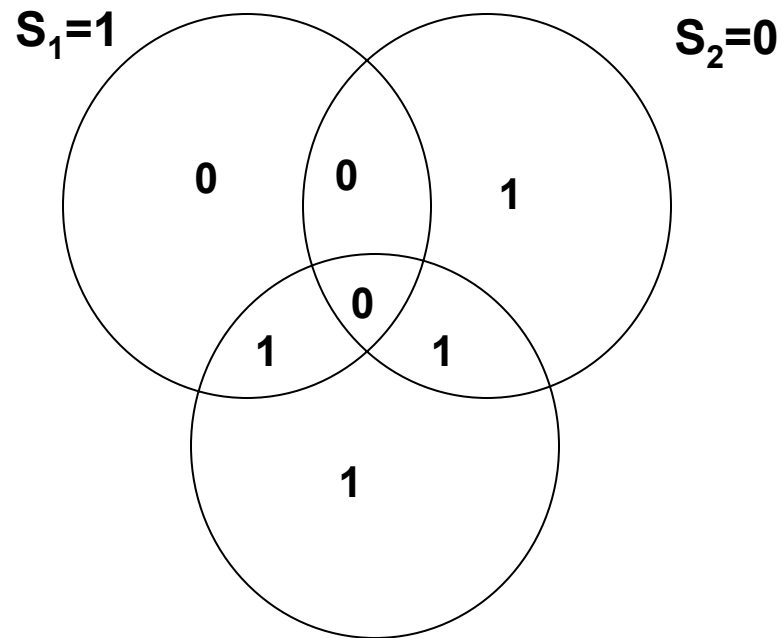
$X=(1110100)$



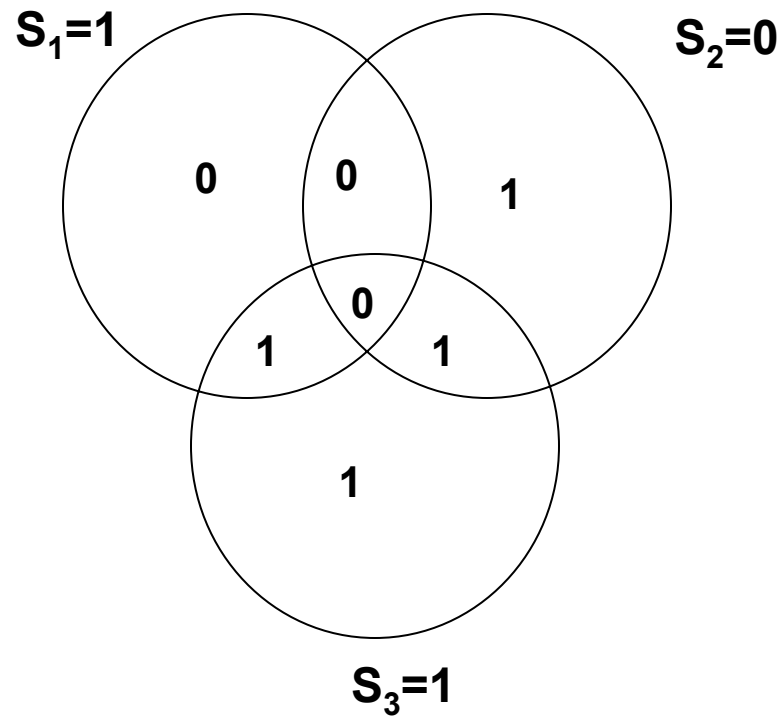
EXPLANATION OF TOY EXAMPLE a la McEliece



EXPLANATION OF TOY EXAMPLE a la McEliece

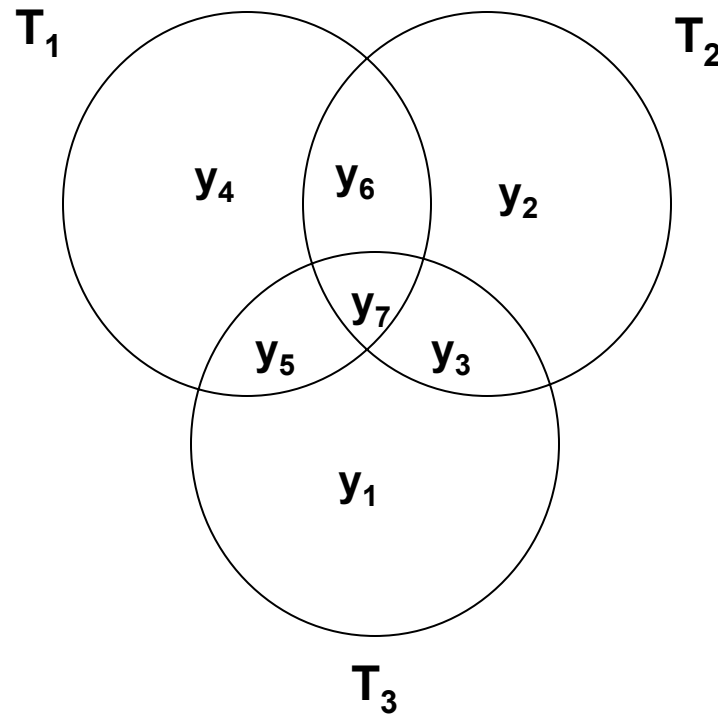


EXPLANATION OF TOY EXAMPLE a la McEliece



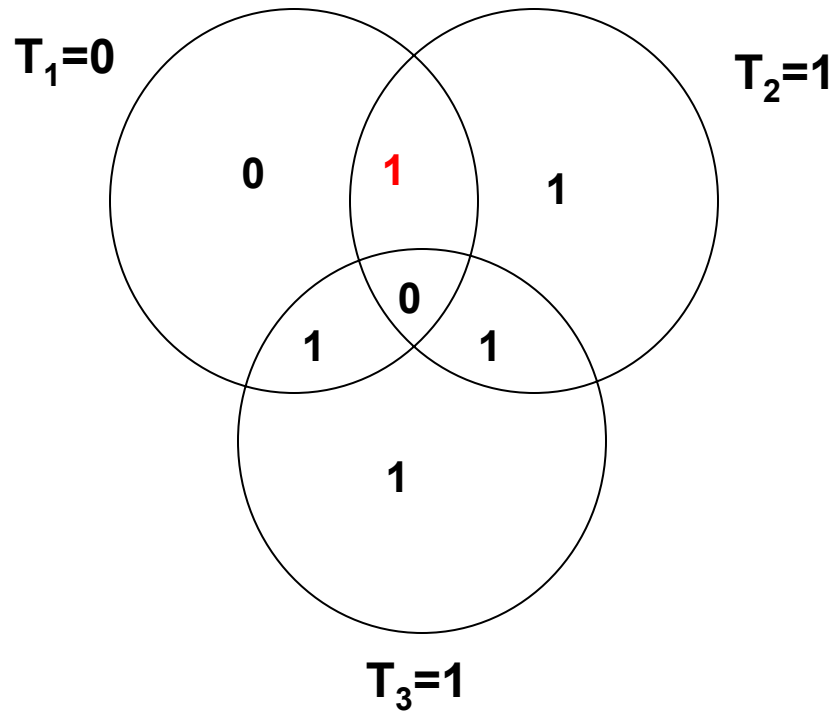
EXPLANATION OF TOY EXAMPLE a la McEliece

Y=(11101**1**0)

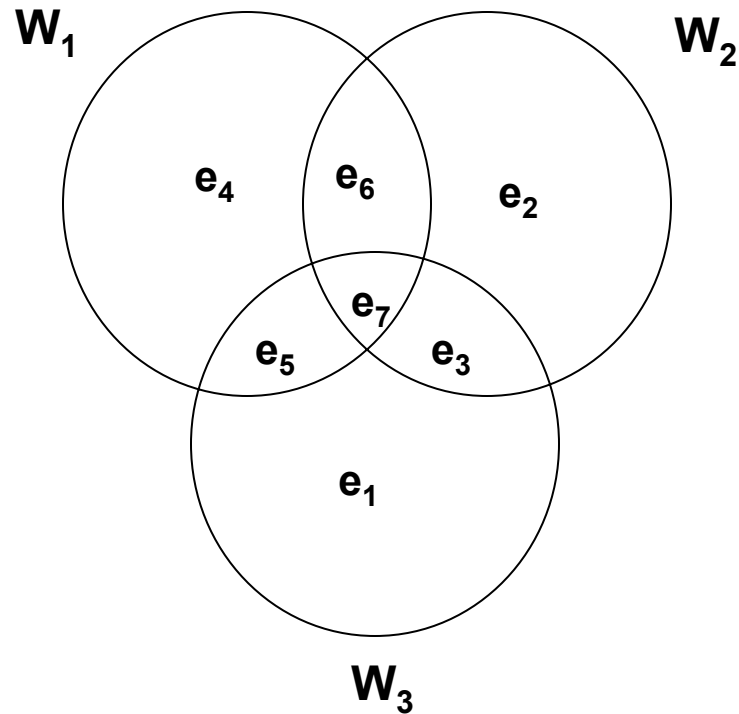


EXPLANATION OF TOY EXAMPLE a la McEliece

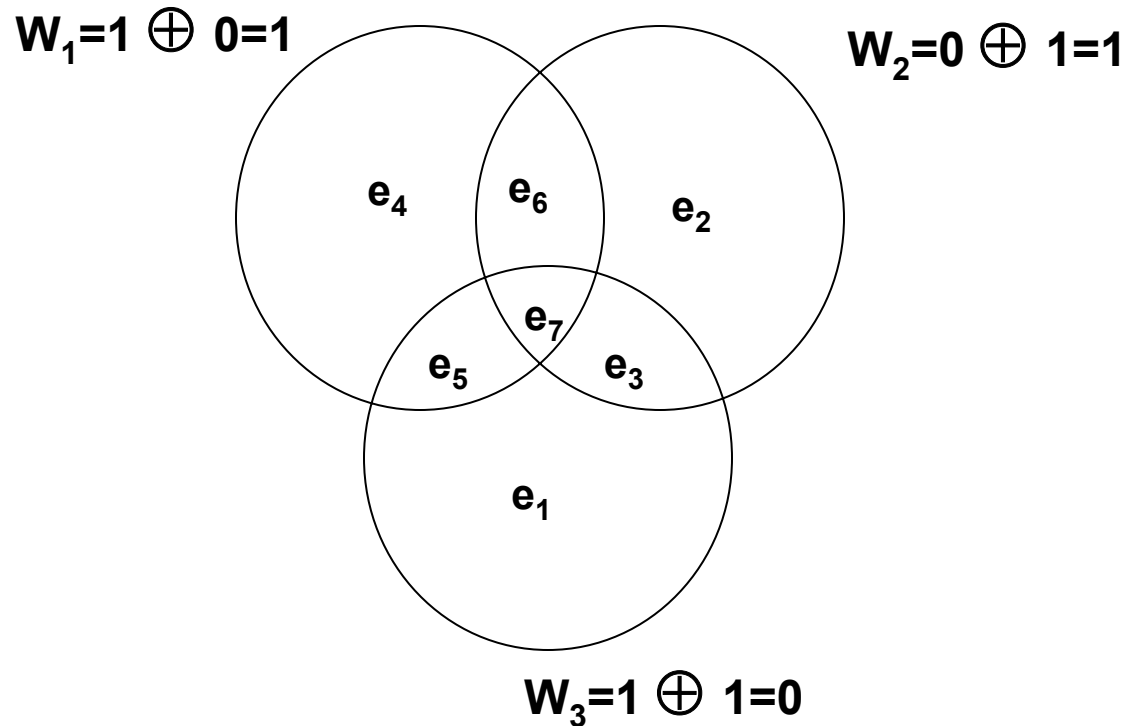
Y=(11101**1**0)



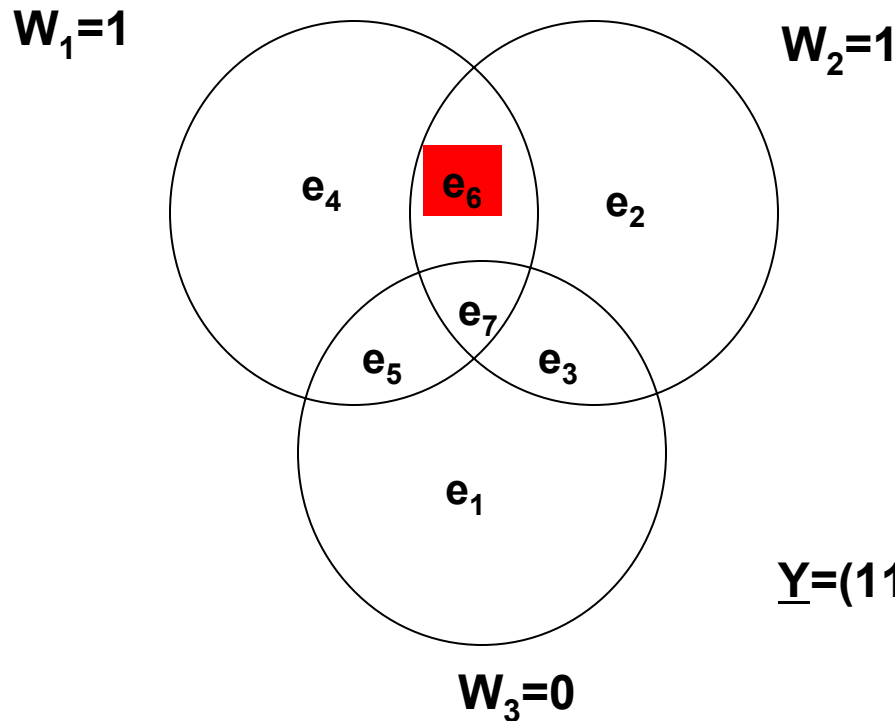
EXPLANATION OF TOY EXAMPLE a la McEliece



EXPLANATION OF TOY EXAMPLE a la McEliece



EXPLANATION OF TOY EXAMPLE a la McEliece

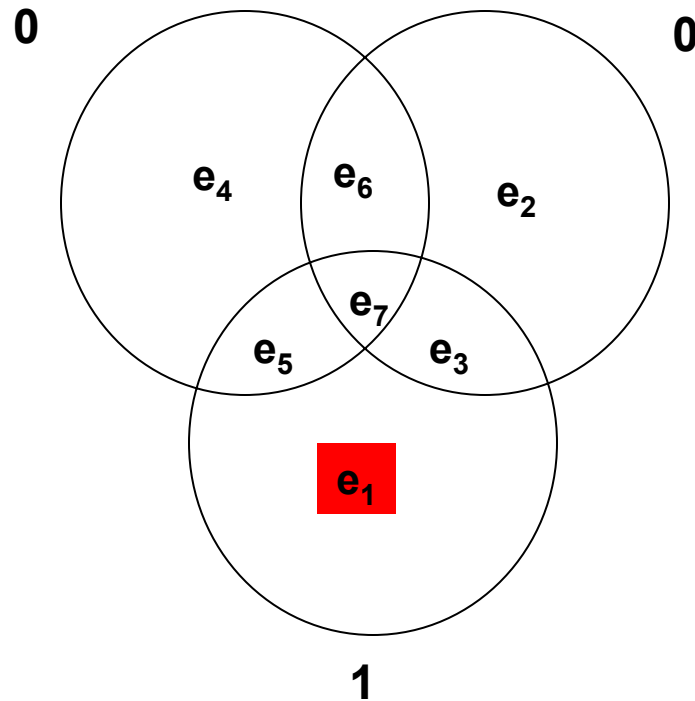


$\underline{Y}=(1110110)$ so $\underline{X}=(1110100)$.

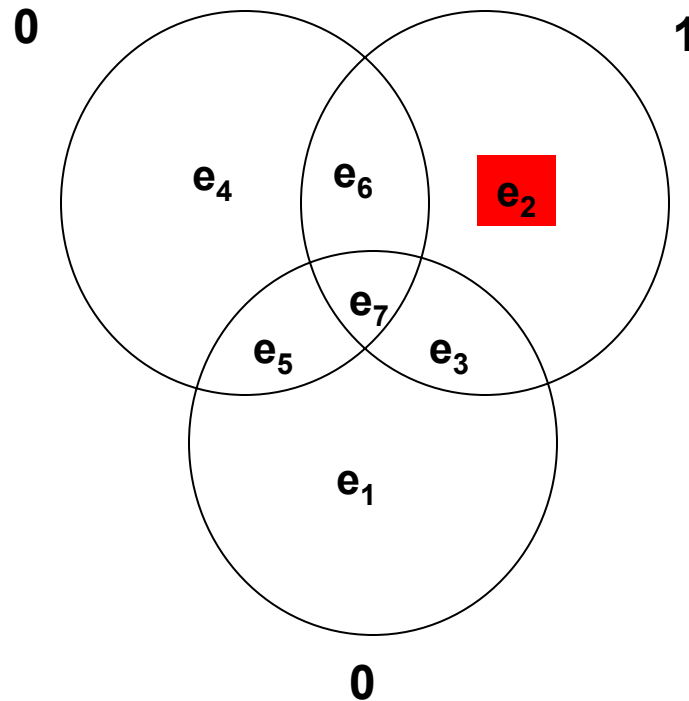
EXPLANATION OF TOY EXAMPLE a la HAMMING

- We have used a (7,4) Hamming single error correcting code.
- We have chosen the parity check matrix so that its columns are the binary representation of the column position. The hash values are the syndromes.
- It should be noted that neither \underline{X} nor \underline{Y} need be a code word.
- Since \underline{S} is the syndrome of \underline{X} and since \underline{T} is the syndrome of \underline{Y} , the composite vector $\underline{W} = \underline{X} \oplus \underline{Y}$ is the syndrome for the **difference** vector $(\underline{X} \oplus \underline{Y})$.

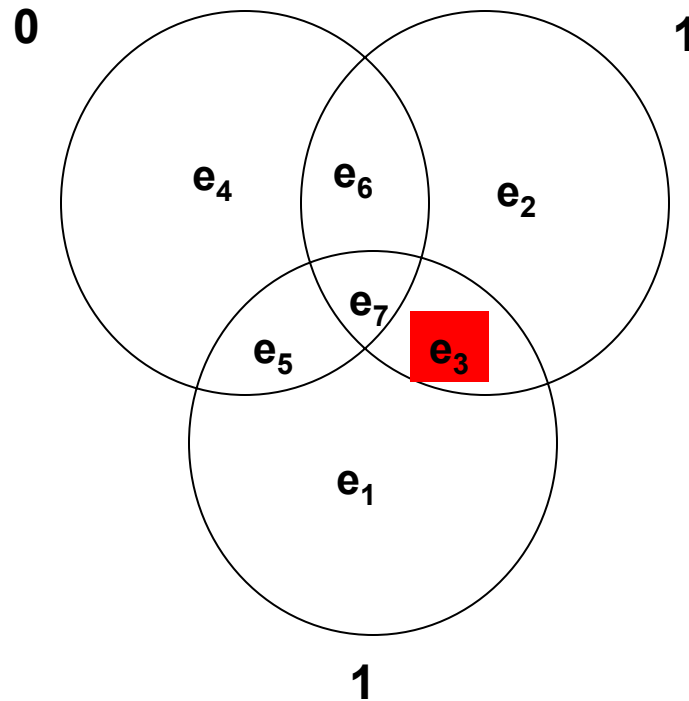
EXPLANATION OF TOY EXAMPLE a la McEliece



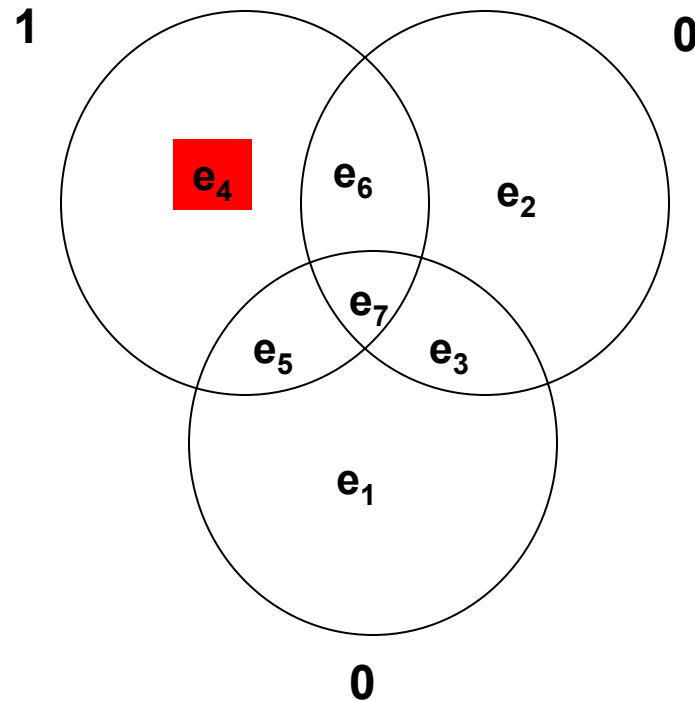
EXPLANATION OF TOY EXAMPLE a la McEliece



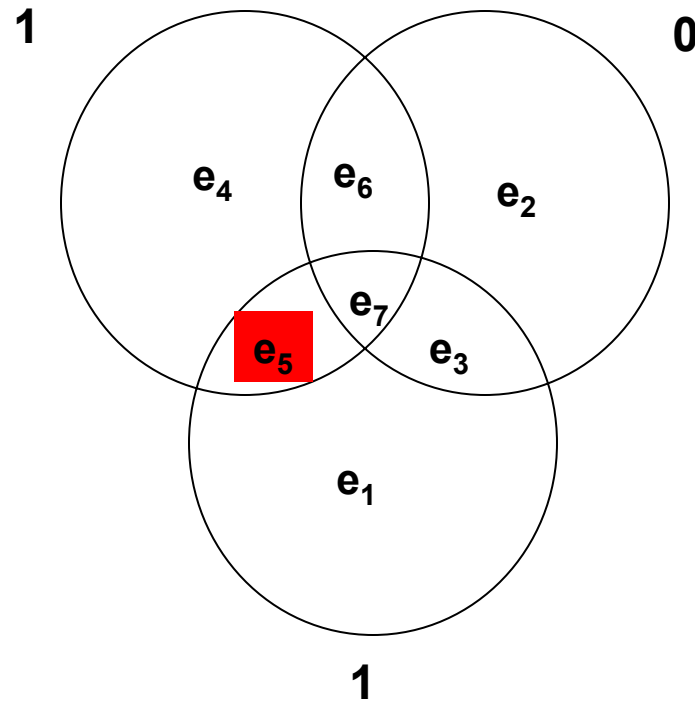
EXPLANATION OF TOY EXAMPLE a la McEliece



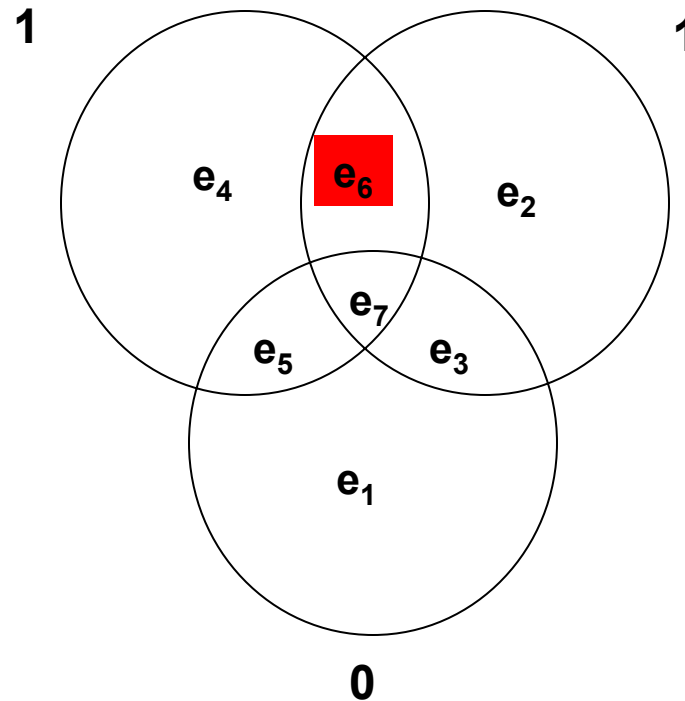
EXPLANATION OF TOY EXAMPLE a la McEliece



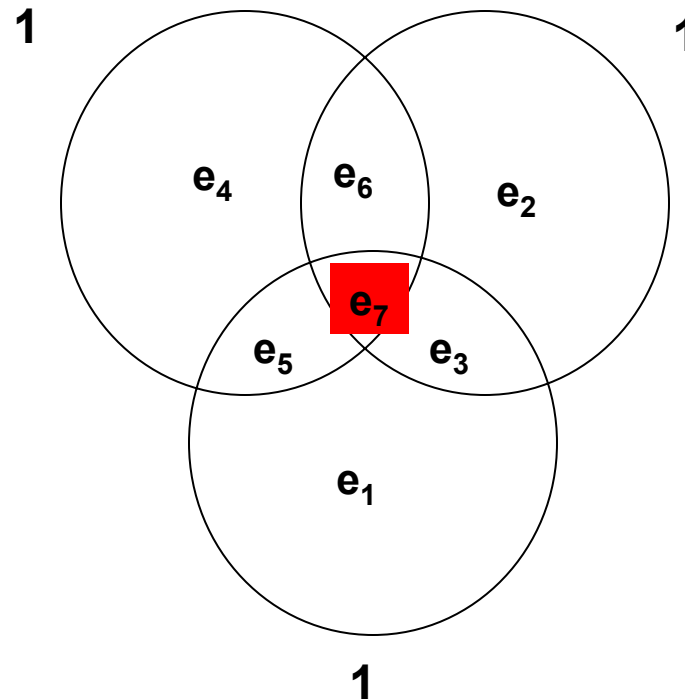
EXPLANATION OF TOY EXAMPLE a la McEliece



EXPLANATION OF TOY EXAMPLE a la McEliece



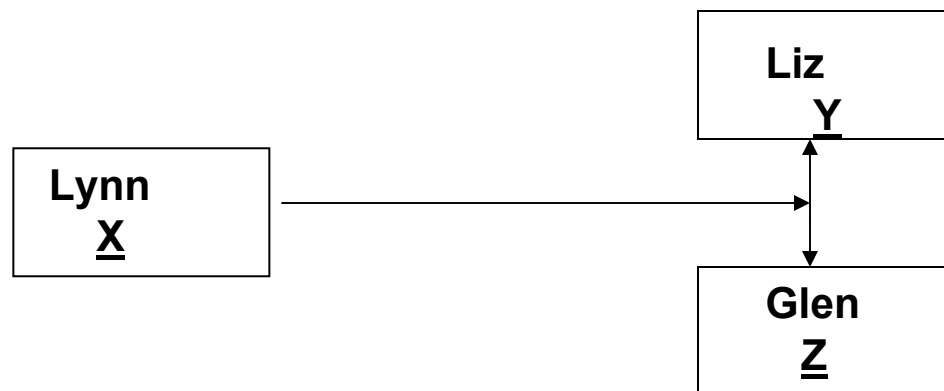
EXPLANATION OF TOY EXAMPLE a la McEliece



TOY EXAMPLE

(2 RECEIVING terminals)

- For the case of one receiving terminal, we have seen that 3 bits of transmission are sufficient, even in the case where Lynn does not know Y.
- We now show that for 2 or more receiving terminals, **3 bits of transmission suffices** even in the case where Lynn does not know either Y or Z.



TOY EXAMPLE

(2 RECEIVING terminals)

- Lynn computes S as before and **broadcasts** S to both Liz and Glen.
- Liz makes the same computation as described previously using her previous knowledge of Y.
- Glen mimics Liz's procedure using Z in place of Y.

TOY EXAMPLE

(ANY NUMBER OF terminals)

- Note that this procedure works for **any number of receiving terminals**.
- That is, 3 bits of information broadcast to all of these terminals allows each one of them to correct their own weather in order to find the weather at the transmitter.
- Note that in this scheme, the transmitter knows only its own weather and not the weather at any of the receivers.

TOY EXAMPLE

- But think about the scheme used when there was only one receiver but when the transmitter knew both X and Y.
- Then the transmitter merely used a **trivial** code to transmit the **difference** between X and Y and the receiver added this difference to Y to obtain X.
- This is **much simpler** than the scheme involving syndromes. (No code was involved!!)
- So, ...

GENERALIZATION

- The previous example described an overly simplistic description as to how the files in the various terminals were related.
- Schemes exists for much more general relationships between the information at the various terminals. In some cases some of the new “almost capacity achieving” codes can be employed (such as turbo codes and LDPC codes).

SOME ANCIENT HISTORY

- The original work was done with David Slepian when I was on sabbatical from P.I.B. at the University of Hawaii.
- The original problem considered was for the binary symmetric source. That is, (X, Y) were i.i.d in pairs where the marginals for both X and Y were equally likely to be 0 and 1 but where $\text{Prob}[X_i = Y_i] = 1-p$.
- Then $H[X] = H[Y] = 1$,
and $H[X|Y] = h(p) = -p \log p - (1-p) \log(1-p)$.

SOME ANCIENT HISTORY

- We originally didn't know whether $H[X|Y]$ could be achieved if the transmitter only knew X .
- After hearing a seminar on coding for the B.S.C, we realized how to do it with a capacity achieving group code and cosets of that group code. The transmitted information identified which coset \underline{X} was in.
- This is equivalent to sending the syndrome of \underline{X} .
- We did not study broadcasting to more than one transmitter.

SOME RECENT HISTORY

- **There has been a revival of interest in this problem as applied to sensor networks.**
- **Furthermore, we now have “almost” capacity achieving codes (e.g., turbo codes and LDPC codes).**
- **A lot more has been done on this problem.**