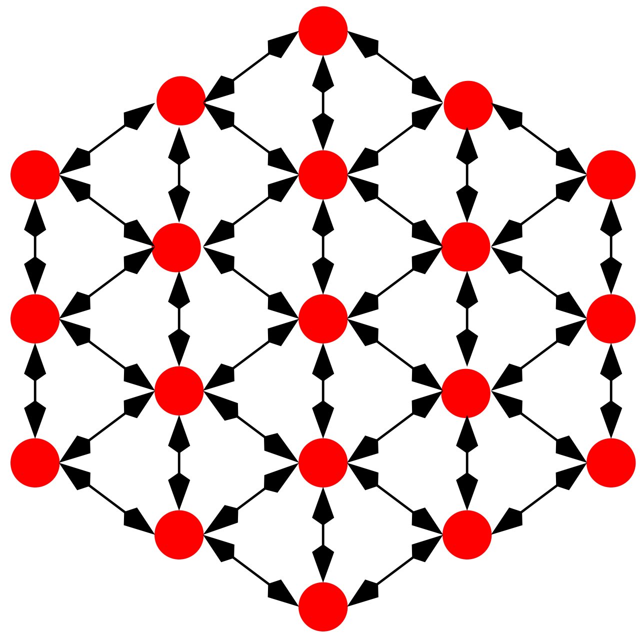
# INFORMATION REPRESENTATION FOR NETWORK SYSTEMS

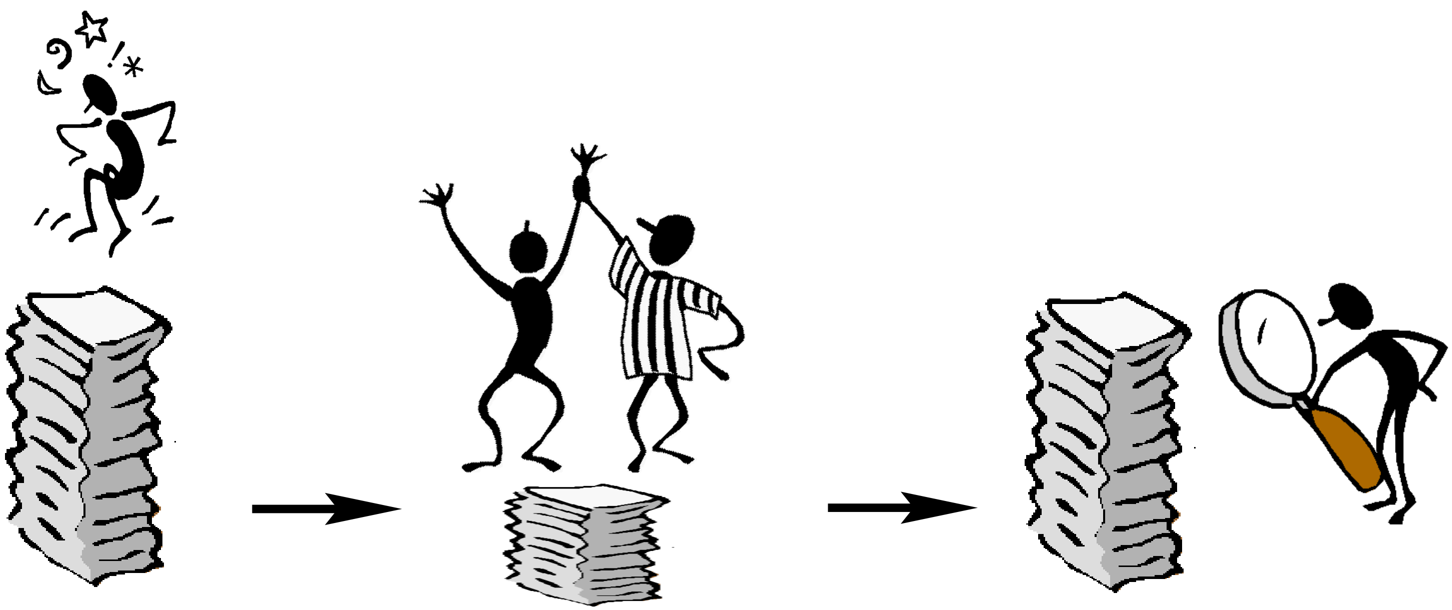


**Michelle Effros**

Department of Electrical Engineering, Lee Center for Advanced Networking
California Institute of Technology

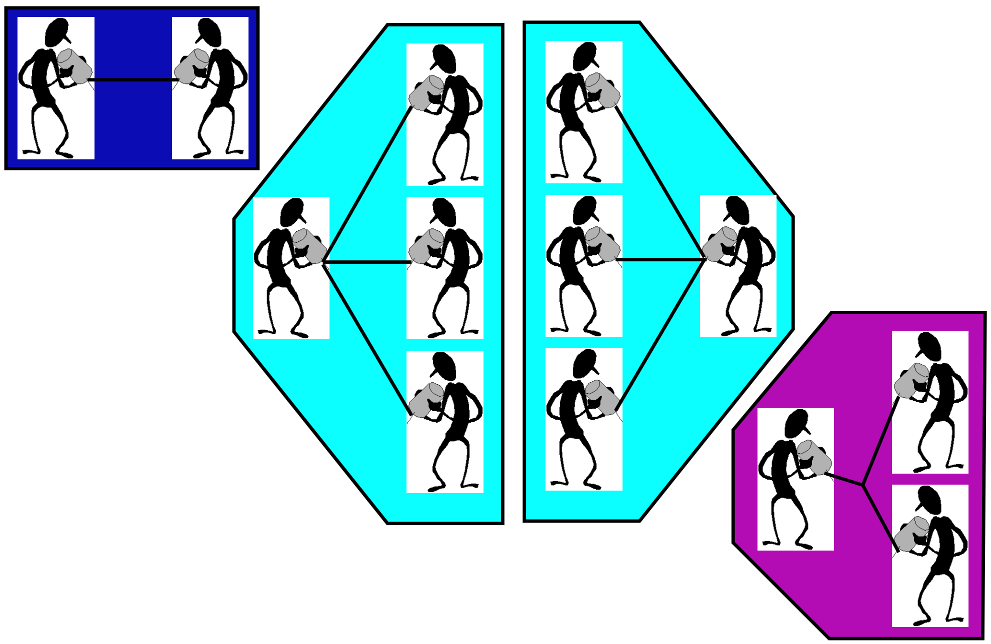
# SETTING THE STAGE

Efficient information representation



## CENTRAL QUESTION:

DOES THE NETWORK MATTER???



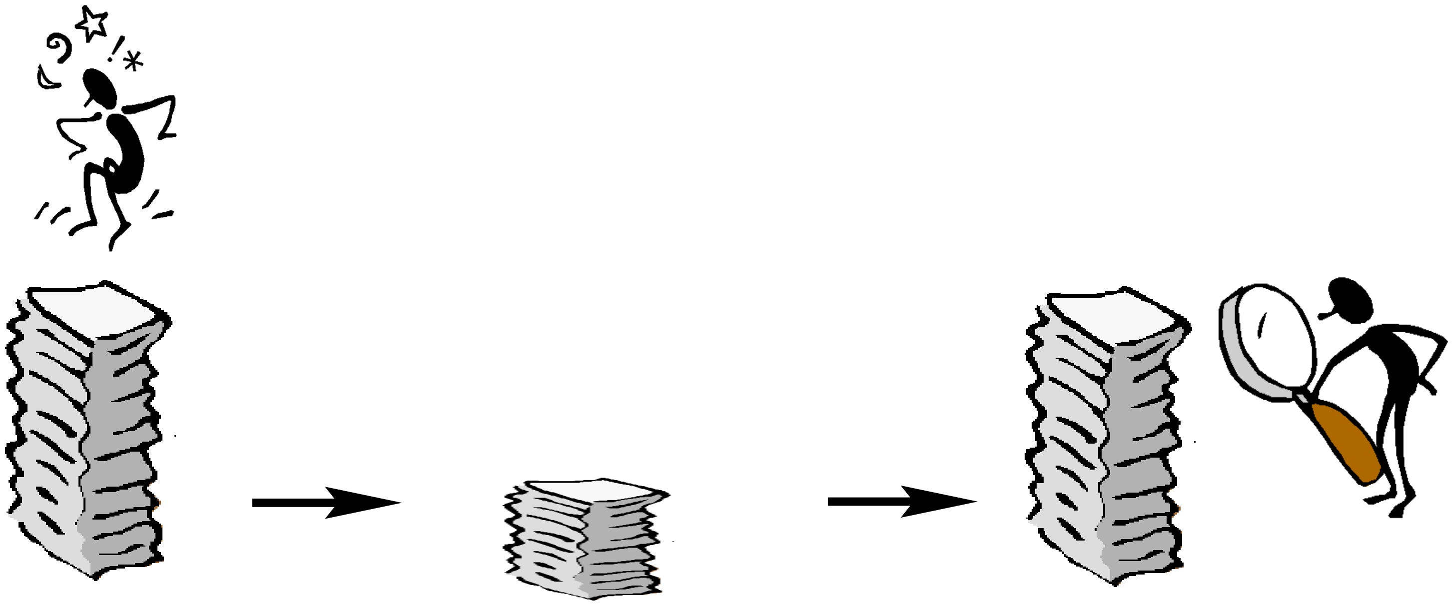
Should the way we represent information change for different network scenarios?

## EXAMPLE:

Our previous compression model implicitly assumed:

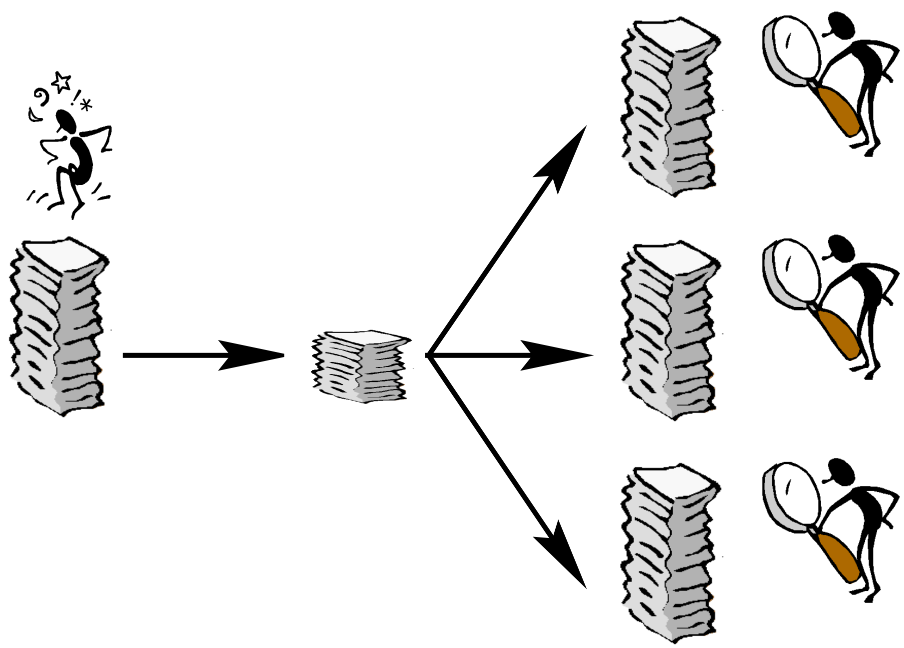
a single sender of information

a single receiver of information



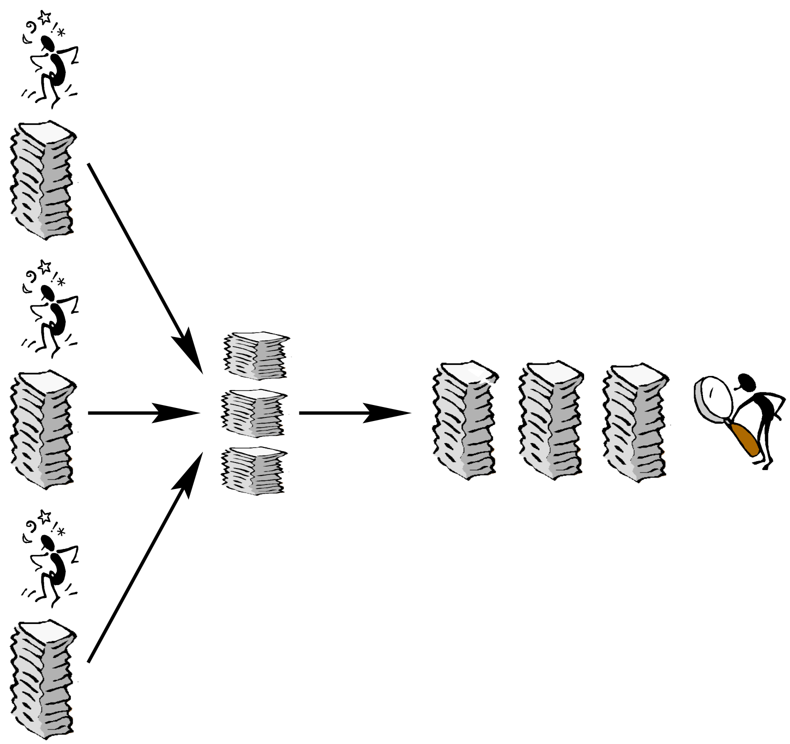
# EXAMPLE:

Would the way we represent information change
if we knew that the data would be used in:
A network with a single sender of information
and multiple receivers of information?



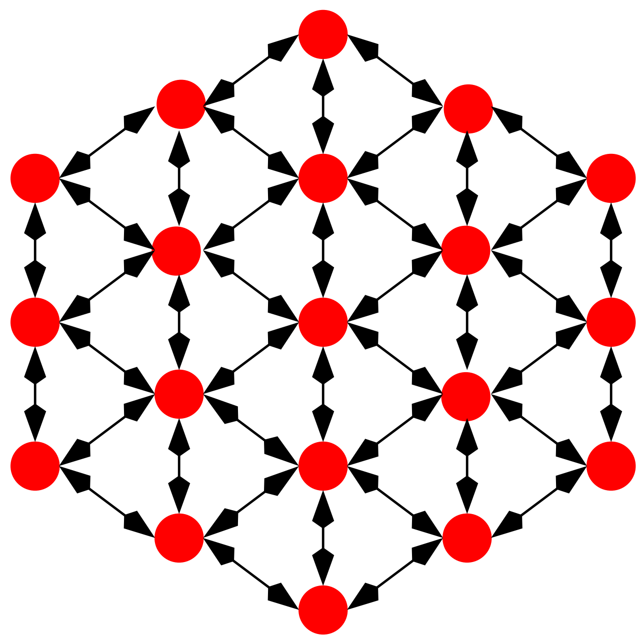
## EXAMPLE:

A network with multiple senders of information and a single receiver of information?



# EXAMPLE:

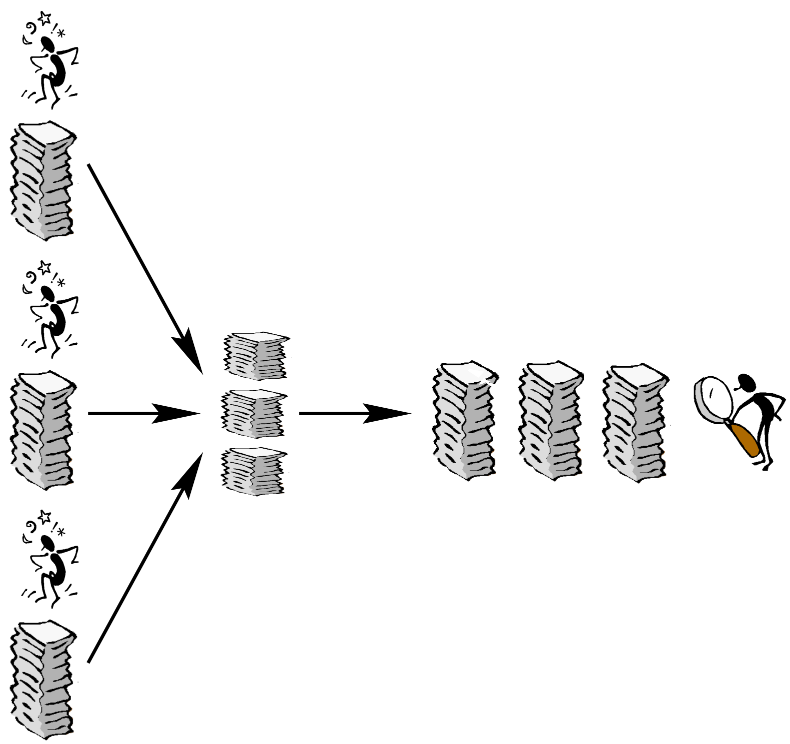
A network with multiple senders of information
and multiple receivers of information?



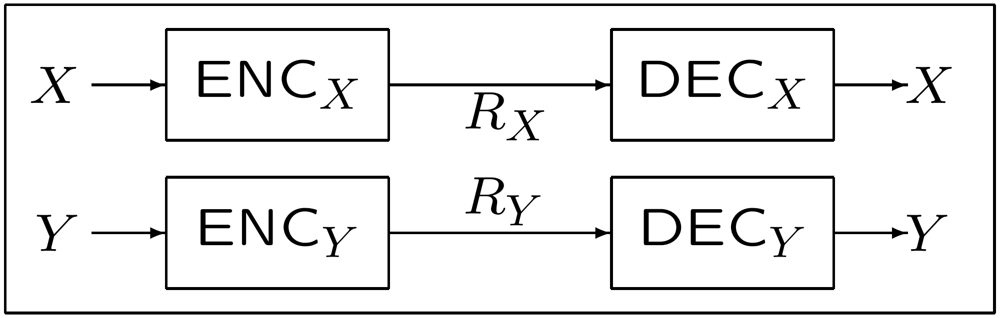
# EXAMPLE: SENSOR NETWORK

Sensors independently collect (correlated) information.

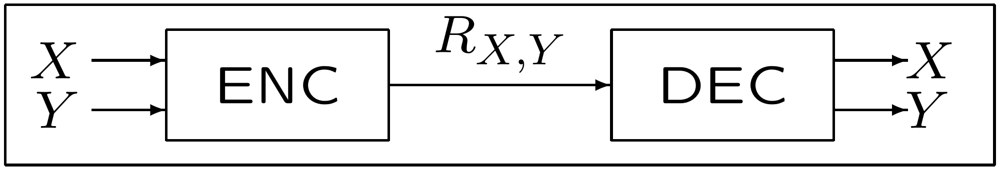
Information compressed & sent to a single base station.



# COMPRESSION FOR SENSOR NETWORKS




Independent Coding




Joint Coding

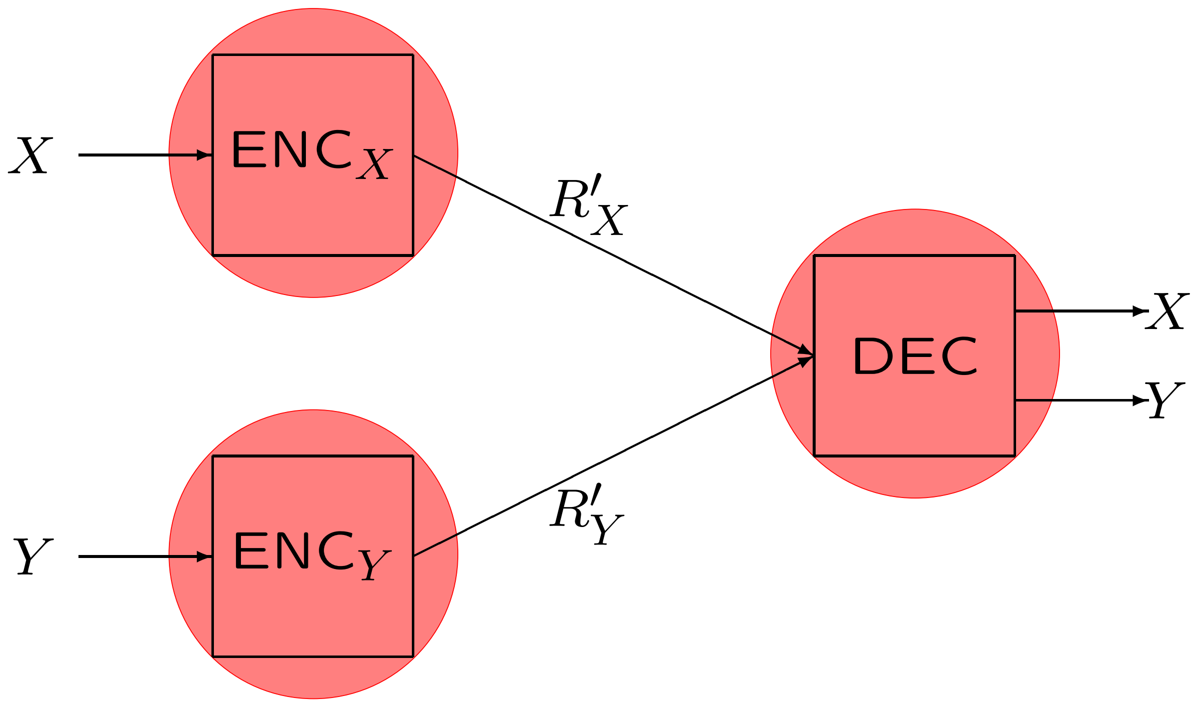
$$R^\star_X + R^\star_Y > R^\star_{X,Y}$$

best possible rate for independent coding $>$ best possible rate for joint coding

*if* $X$ and $Y$ are *not* independent.

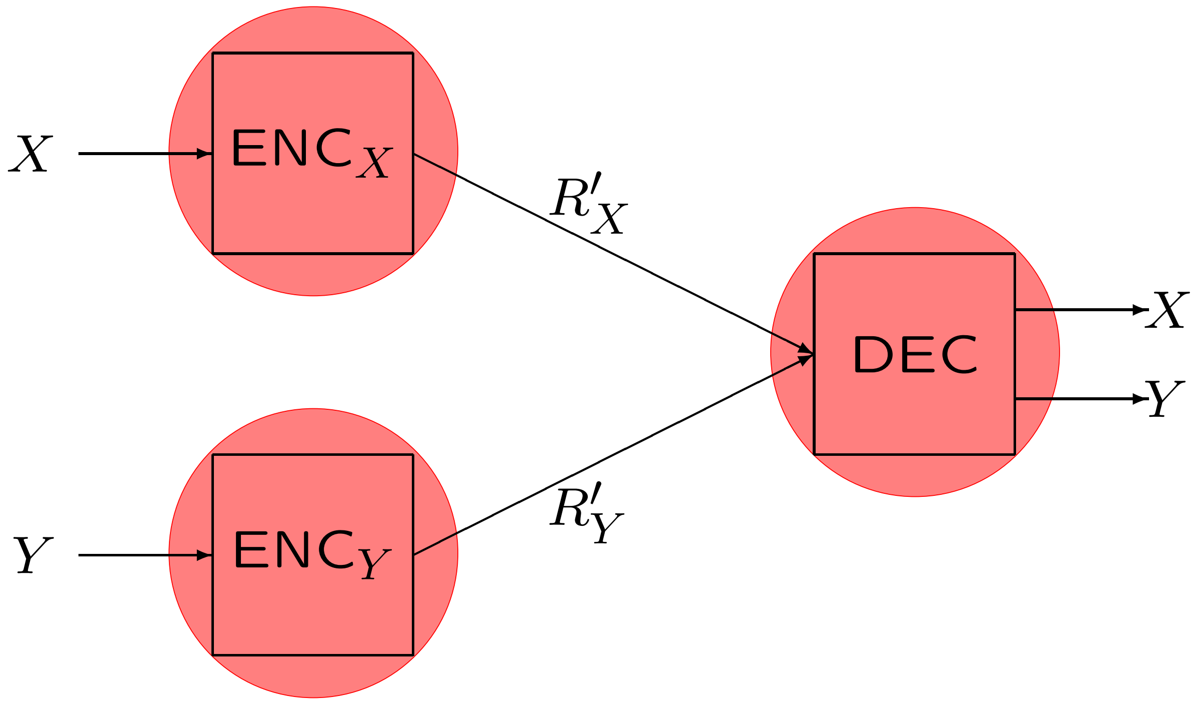
# MULTIPLE ACCESS DATA REPRESENTATION

The closest our sensor network can come to joint coding is:




Is joint decoding sufficient
to achieve the performance of the joint code?

# MULTIPLE ACCESS DATA REPRESENTATION




Slepian and Wolf (1973) showed:

$R'_X + R'_Y = R^{\star}_{X,Y}$ is (theoretically) achievable!

(with coding dimension $n \to \infty$ and probability of error $P_e \to 0$)

# MULTIPLE ACCESS DATA REPRESENTATION

BUT today's products still do independent coding!

Practical multiple access code design has remained an unsolved problem for almost 30 years.

# WHY?

- Many techniques from independent coding fail when applied to network codes (e.g., instantaneous $\not\Leftrightarrow$ prefix-free)

- Rates achievable by near-lossless codes ($P_e < \epsilon$ for any $\epsilon > 0$) are not necessarily achievable by lossless codes ($P_e = 0$).

# INSTANTANEOUS vs. PREFIX-FREE

**Traditional lossless data compression:**

- Each symbol is represented by a string of bits.
- The table of all symbols & binary strings describes a code.
  Example:

| CODE 1 | |
|---|---|
| Symbol | Binary string |
| $a$ | 00 |
| $b$ | 01 |
| $c$ | 10 |
| $d$ | 11 |

$$\text{Code}_1(cabaa) = 1000010000$$

# INSTANTANEOUS vs. PREFIX-FREE

**Traditional lossless data compression:**

- To achieve compression, use:
  - short strings for symbols that appear frequently
  - long strings for symbols that appear infrequently

  Example:

| Symbol | Code 1 | Code 2 |
|---|---|---|
| $a$ | 00 | 0 |
| $b$ | 01 | 10 |
| $c$ | 10 | 110 |
| $d$ | 11 | 111 |

$$\text{Code}_1(cabaa) = 1000010000$$
$$\text{Code}_2(cabaa) = 11001000$$

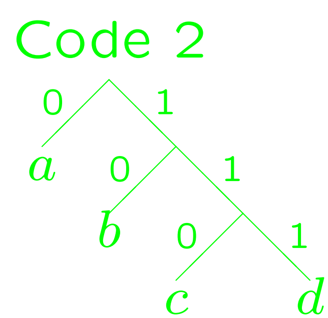
## INSTANTANEOUS vs. PREFIX-FREE

**Traditional lossless data compression:**

- We further require instantaneous decoding ⇔
  - Unique decodability (to make the code lossless).
  - Immediate decodability (to keep delay low).

Example:

| Symbol | Code 3 | Code 2 |
|---|---|---|
| $a$ | 0 | 0 |
| $b$ | 10 | 10 |
| $c$ | 010 | 110 |
| $d$ | 011 | 111 |



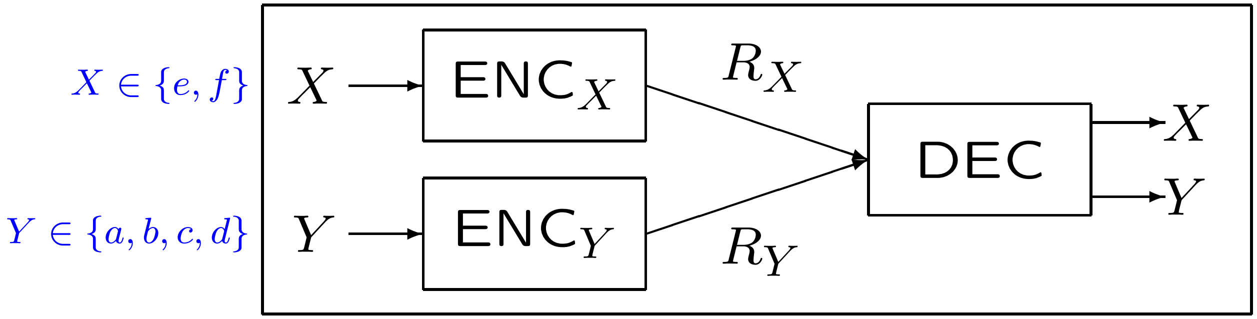

Code 3: not immediate (0 $\Rightarrow a$? or $b$? or $c$?)

not uniquely decodable (Code$_3(ab)$ = Code$_3(c)$ = 010)

Code 2: prefix-free ⇔ instantaneous (for traditional codes)

## INSTANTANEOUS vs. PREFIX-FREE

**Multiple-access lossless data compression:**



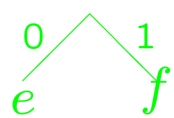

Prefix-free $\not\Leftrightarrow$ instantaneous

# INSTANTANEOUS vs. PREFIX-FREE
## Multiple-access lossless data compression:

Prefix-free $\not\Leftrightarrow$ instantaneous

Example: The multiple-access code $(\text{Code}_X, \text{Code}_Y)$

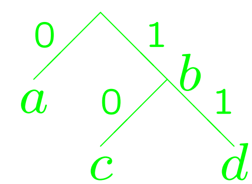
Code$_X$




| Symbol | Code$_X$ |
|---|---|
| $e$ | 0 |
| $f$ | 1 |

| Symbol | Code$_Y$ |
|---|---|
| $a$ | 0 |
| $b$ | 1 |
| $c$ | 10 |
| $d$ | 11 |

Code$_Y$




is uniquely and immediately decodable for distribution $p(x, y)$.

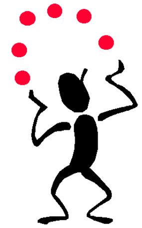
| $p(x,y)$ | | | | |
|---|---|---|---|---|
| $X \backslash Y$ | $a$ | $b$ | $c$ | $d$ |
| $e$ | .35 | .10 | 0 | 0 |
| $f$ | .45 | 0 | .05 | .05 |

## WHY?

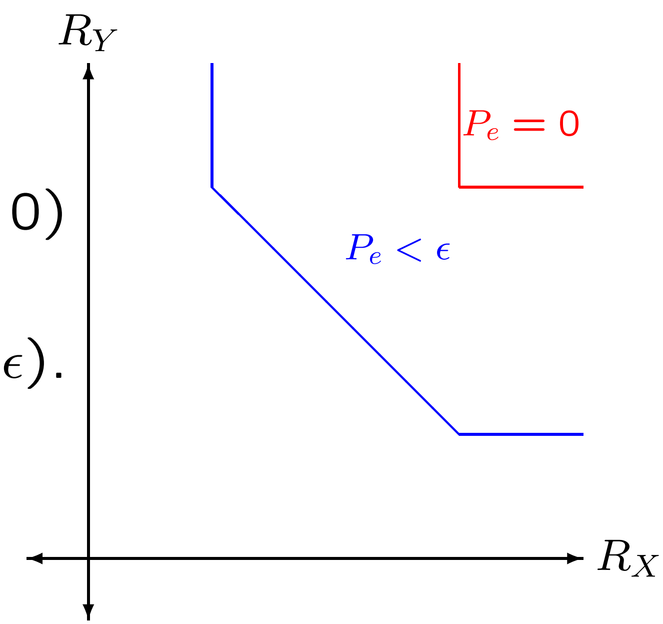
- Many techniques from independent coding fail when applied to network codes (e.g., instantaneous $\not\Leftrightarrow$ prefix-free)

- Rates achievable by near-lossless codes ($P_e < \epsilon$ for any $\epsilon > 0$) are not necessarily achievable by lossless codes ($P_e = 0$).

## $P_e = 0$ vs. $P_e < \epsilon$



**Multiple-access lossless data compression:**



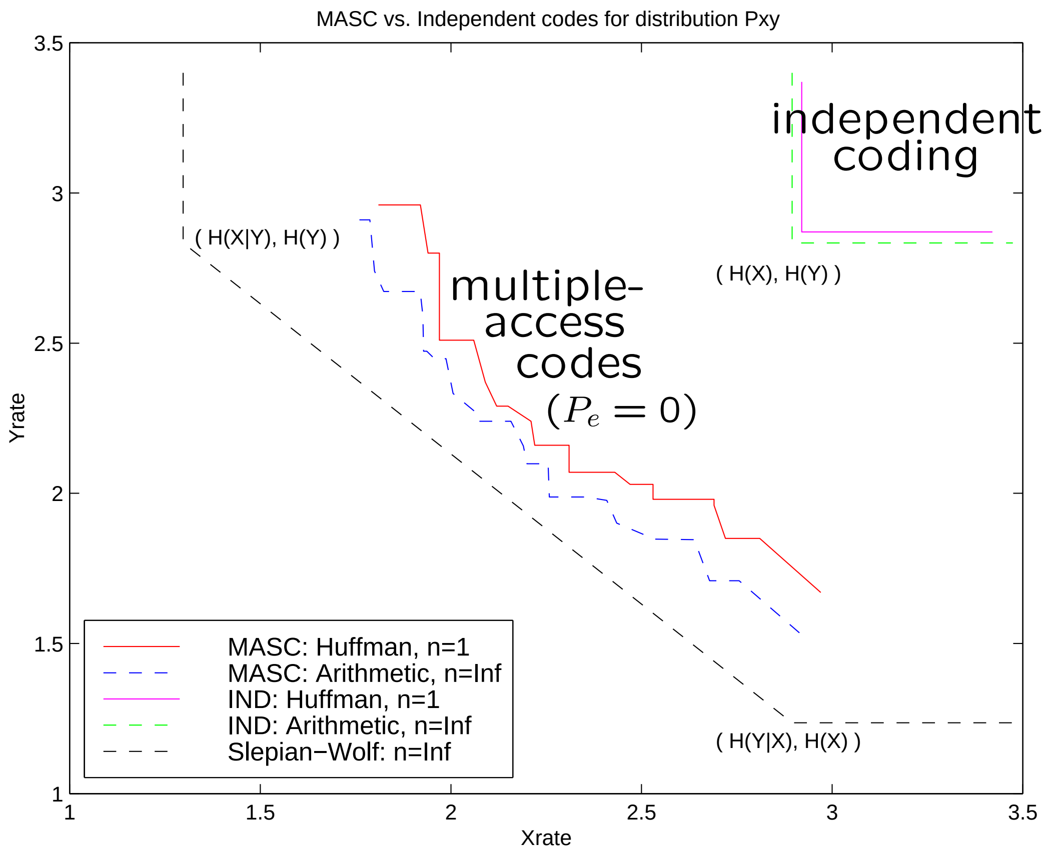

- There exist source distributions $p(x, y)$ such that for any $\epsilon > 0$:
  (the lowest rates achievable with $P_e = 0$)
  $$\gg$$
  (the lowest rates achievable with $P_e < \epsilon$).
- $P_e = 0$ is required for lossless coding.
- $P_e < \epsilon$ suffices within lossy codes.
- The rates achievable at each $P_e$ vary with the code dimension $n$.
- We wish to find the optimal code for arbitrary $P_e$ and $n$.
  ($\sim$ 30 year old unsolved problem...)

# SOLUTIONS

- [Zhao & Effros - DCC 2001], [Zhao & Effros - ISIT 2001] [Zhao & Effros - 2001] introduce optimal algorithms for designing lossless ($P_e = 0$) and near-lossless ($P_e < \epsilon$ for arbitrary $\epsilon$) data compression algorithms for multiple access networks.
- [Fleming & Effros - DCC 2001],[Fleming, Zhao & Effros - 2001] introduce an optimal algorithm for designing lossy codes for arbitrary networks.

See the poster session for more details!

# RESULTS



MASC vs. Independent codes for distribution Pxy
3.5
3
2.5
2
1.5
1
Yrate
1
1.5
2
2.5
3
3.5
Xrate
( H(X|Y), H(Y) )
( H(X), H(Y) )
( H(Y|X), H(X) )
independent coding
multiple-access codes
$(P_e = 0)$
MASC: Huffman, n=1
MASC: Arithmetic, n=Inf
IND: Huffman, n=1
IND: Arithmetic, n=Inf
Slepian−Wolf: n=Inf

# CONCLUSIONS

Information representation for networks requires
fundamentally new techniques
to address the:

- Multiple users
- High connectivity
- Heterogeneity
- Time-variance

of modern network applications.

Developing these new paradigms yields enormous improvements in system:

- Performance
- Functionality.