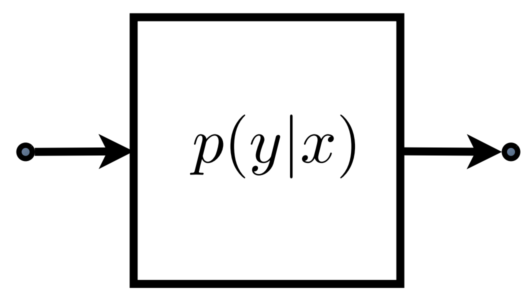
On Networks, Capacities, and Controls

Michelle Effros
California Institute of Technology

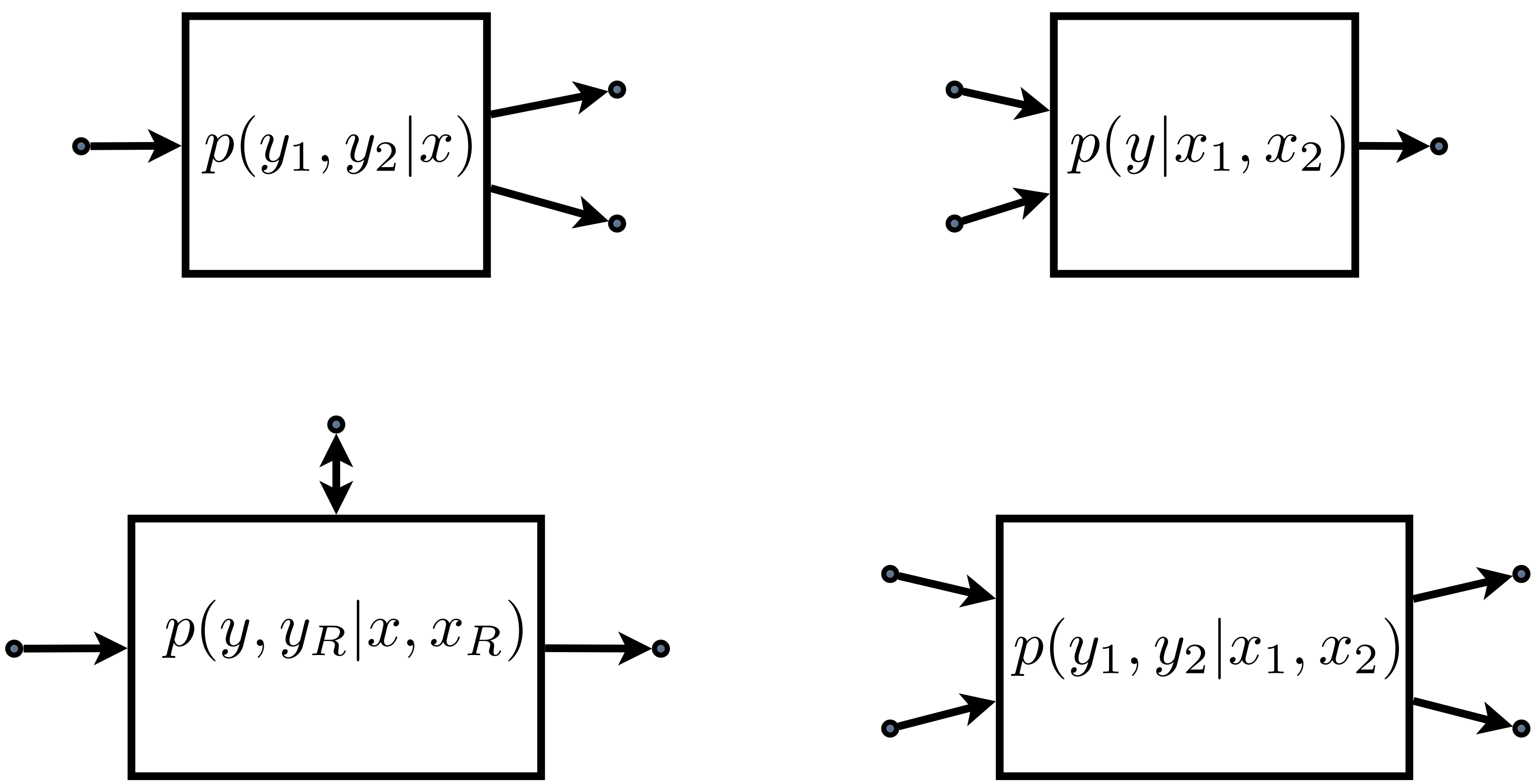
Information theory began with channel capacity.



$$C = \max_{p(x)} I(X;Y)$$

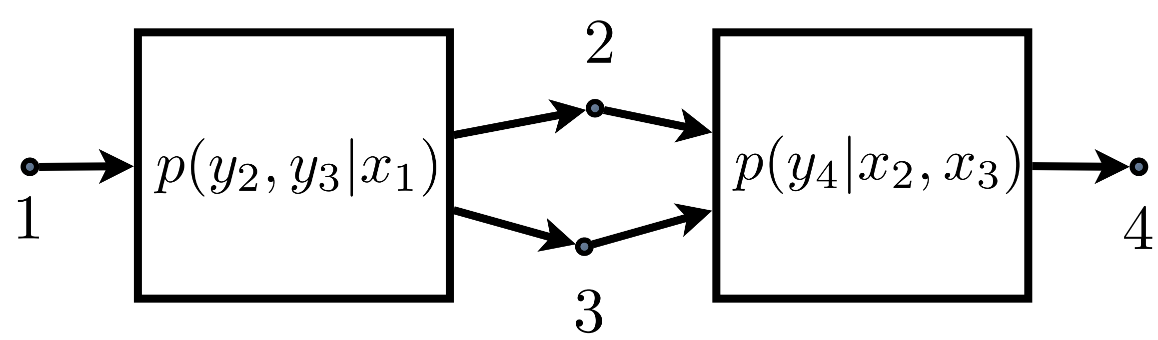
[Shannon, 1948]

Today's focus is largely on network capacities.



Deriving network capacities is challenging.
Partial solutions are available for ALL of these networks.
Complete solutions are not available for ANY of these networks.

Unfortunately, capacities do not compose.

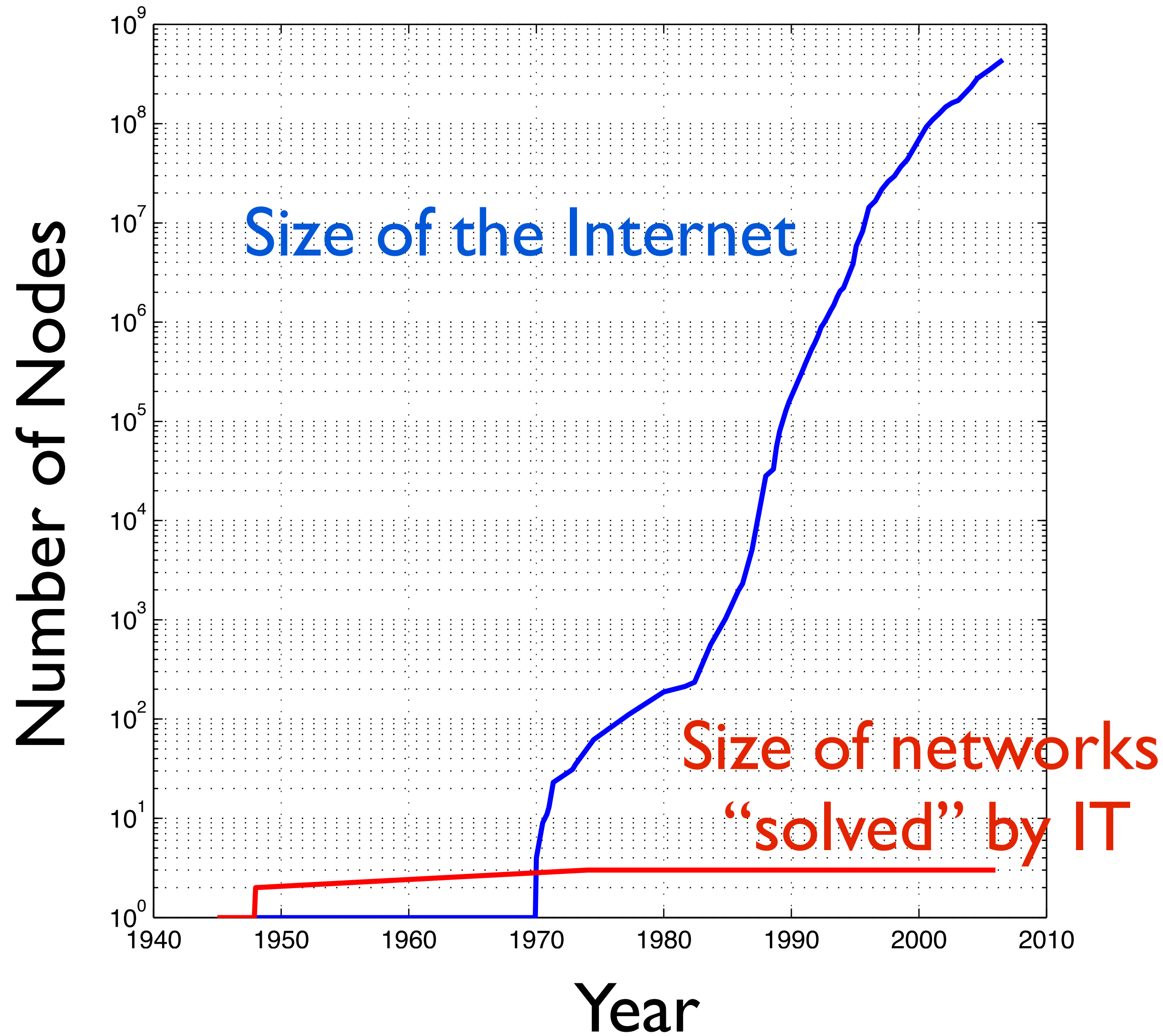


EXAMPLE:

The capacity of the diamond network can be MUCH larger than the maximal sum-rate through each channel.

$$\max R_{1\to 4} >> \max(R_{1\to 2} + R_{1\to 3} + R_{1\to\{2,3\}}),$$
$$\max R_{1\to 4} >> \max(R_{2\to 3} + R_{3\to 4})$$

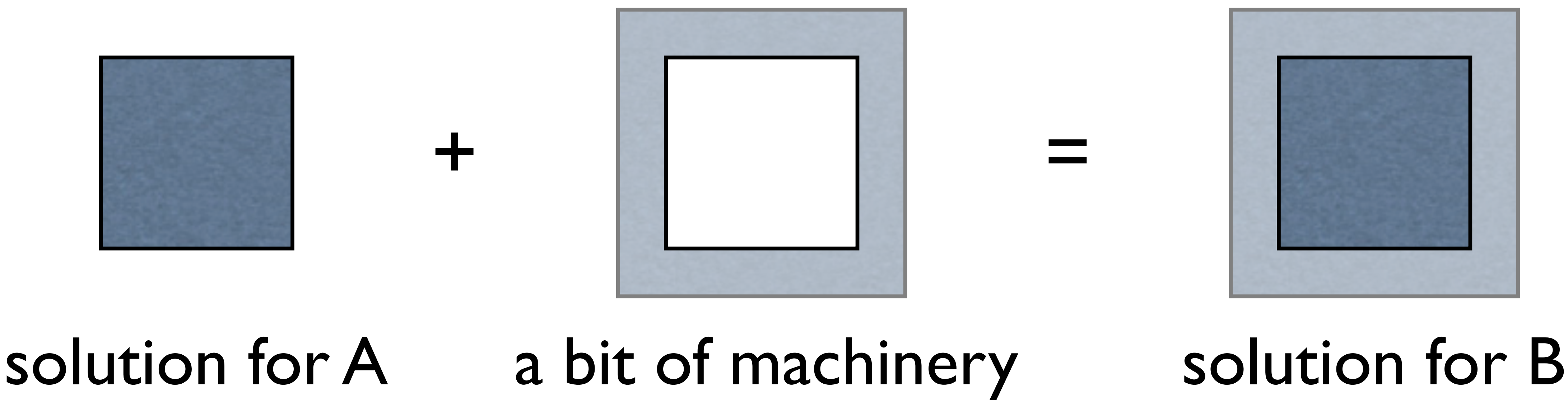
The gap between theory and practice is widening.



Michelle Effros, LCCC 2012

Reduction is a great tool for solving hard problems.

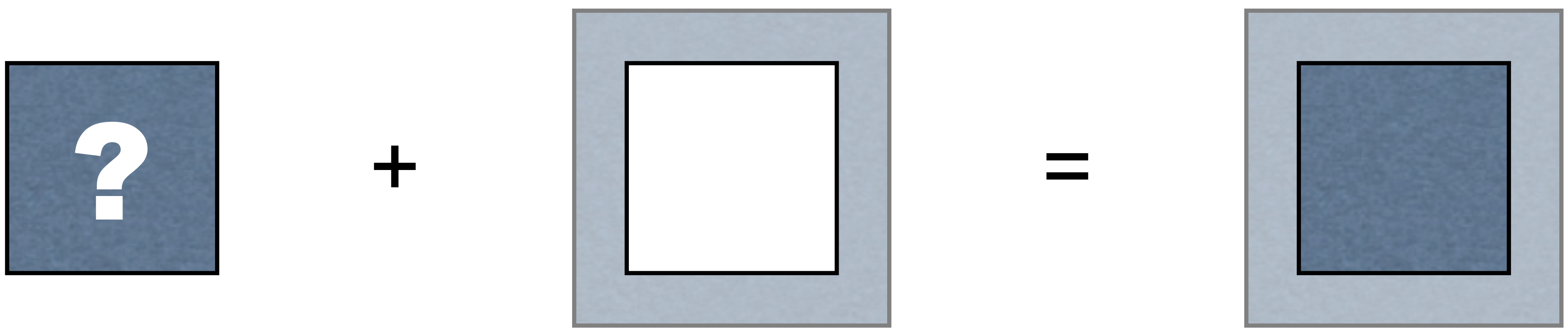
Given: Problems A & B



If the "machinery" is simple,
then B can't be much harder than A.

Reduction is a great tool for ~~solving~~ AVOIDING hard problems.

Given: Problems A & B



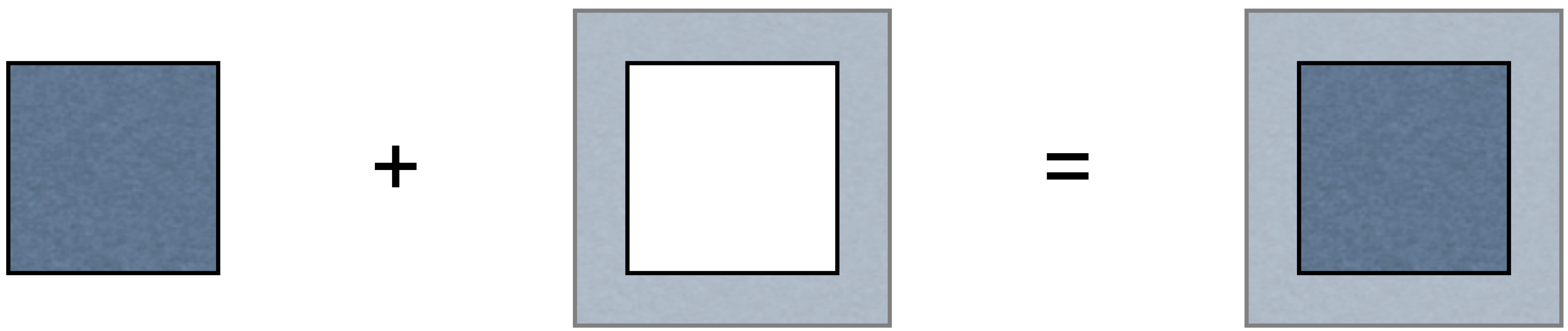
solution for A | a bit of machinery | solution for B

The relationship follows even if no solution for A is known.

It is enough to build the "machinery."

The same strategy applies in information theory.

Given: Networks A & B

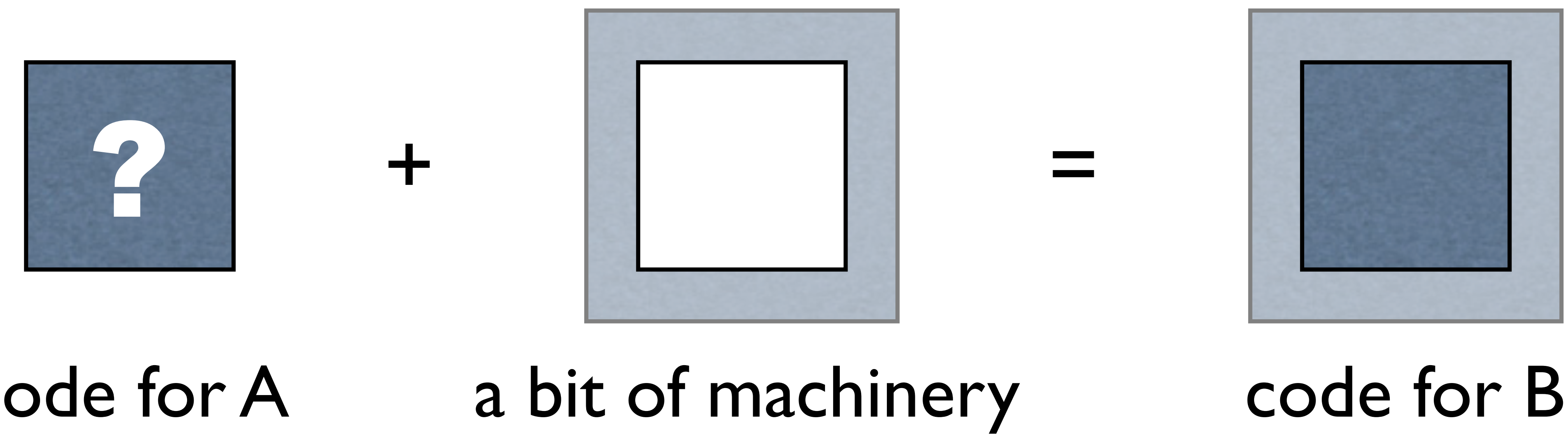


code for A   a bit of machinery   code for B

If the machinery (asymptotically) guarantees the same performance (error probability & rate), then any rate achievable on A is achievable on B.

The same strategy applies in information theory.

Given: Networks A & B

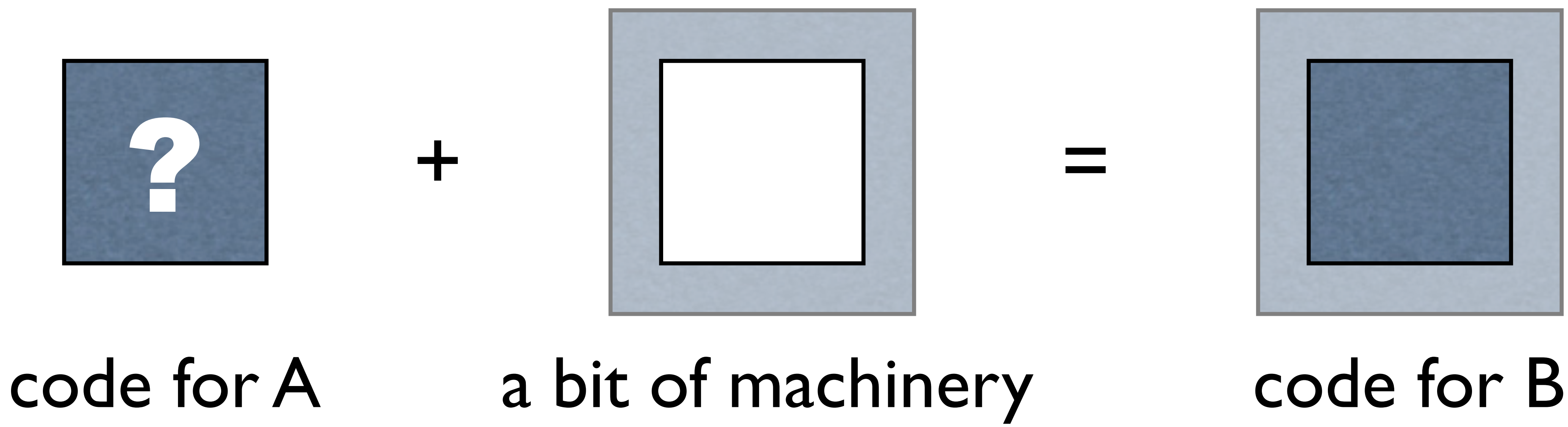


The relationship holds even if the code for A is absent.

All we need is the “machinery.”

The same strategy applies in information theory.

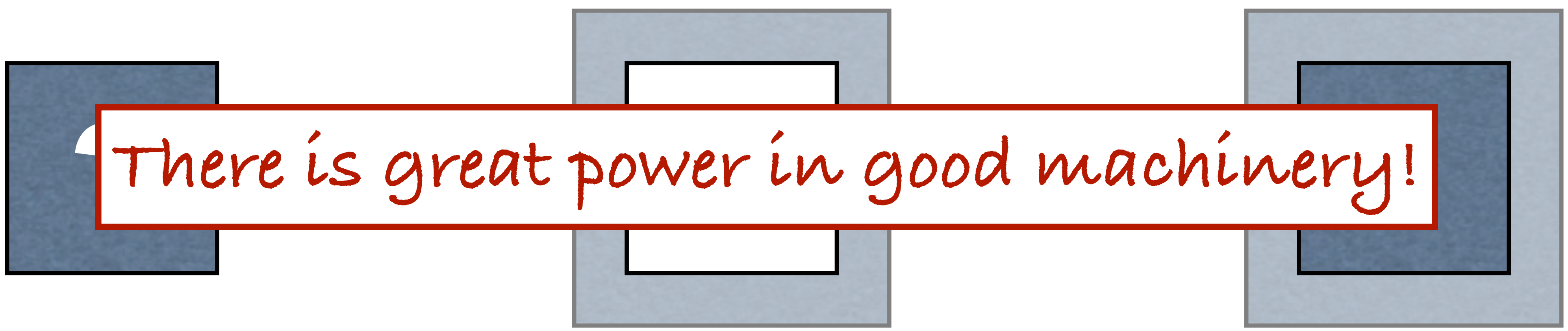
Given: Networks A & B



Proving Capacity(A) is a subset of Capacity(B)
requires no codes and no knowledge of either capacity.

The same strategy applies in information theory.

Given: Networks A & B

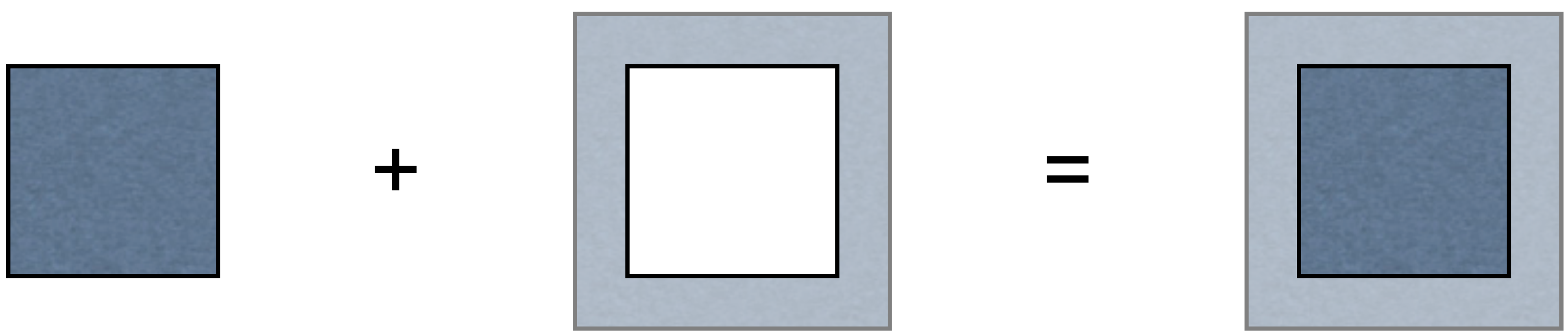


code for A — a bit of machinery — code for B

Proving Capacity(A) is a subset of Capacity(B) requires no codes and no knowledge of either capacity.

This strategy is not new.

Given: A & B



For example...

CS Theory [Hartmanis & Stearns, 1965]
Info Theory [Slepian & Wolf, 1973]

Outline

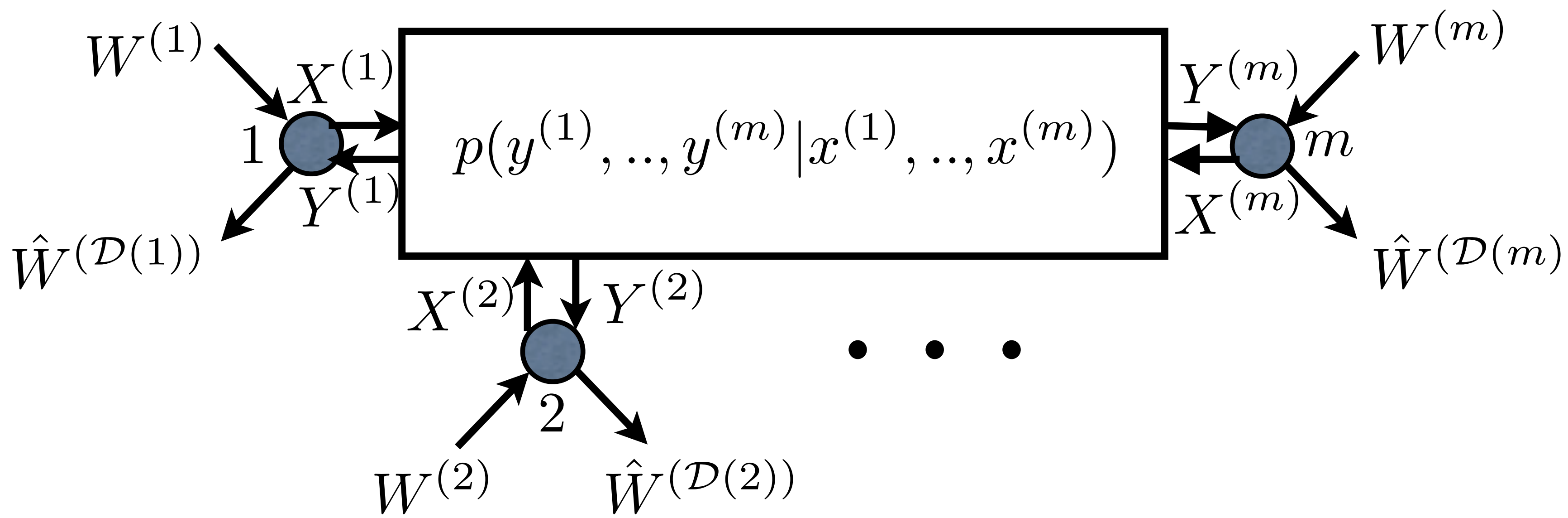
Definitions

How does delay affect network capacity?

Is there a path to a scalable information theory?

Can we move beyond capacity to controls?

Consider a memoryless m-node network.



$$\begin{aligned}
\mathcal{N} &= m\text{-node memoryless network,} \\
W^{(i)} &= \text{independent message originating at node } i \\
\mathcal{D}(i) &\subset \{1, \ldots, m\} = \text{messages required at node } i
\end{aligned}$$

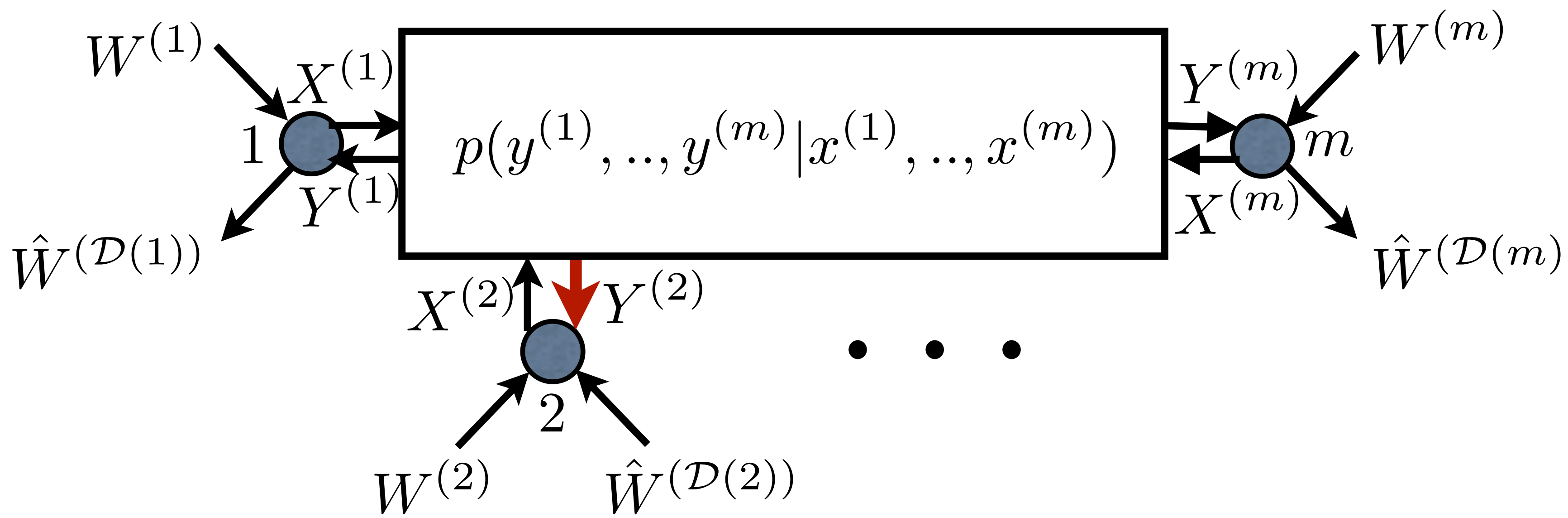
For a blocklength-n code,

$$\begin{aligned}
W^{(i)} &\in \{1, \ldots, 2^{nR^{(i)}}\} \\
(X_t^{(i)}, Y_t^{(i)}) &= \text{network (input,output) of node i at time } t \in \{1, \ldots, n\} \\
X_t^{(i)} &= X_t^{(i)}(W^{(i)}, Y_{1:t-1}^{(i)}) \quad \text{(causality constraint)} \\
\hat{W}^{(\mathcal{D}(i))} &= (\hat{W}^{(j)} : j \in \mathcal{D}(i)) = \hat{W}^{(\mathcal{D}(i))}(W^{(i)}, Y_{1:n}^{(i)}) \\
&= \text{reconstruction at node } i \text{ after } n \text{ time steps} \\
P_e^{(n)} &= \Pr(\textstyle\bigcup_{i \in \{1,\ldots,m\}} \{\hat{W}^{(D(i))} \neq (W^{(j)} : j \in \mathcal{D}(i))\})
\end{aligned}$$

Network capacity:

$$\text{Capacity}(\mathcal{N}) = \overline{\{(R^{(1)}, \ldots, R^{(m)}) : \exists \text{ seq of codes with } P_e^{(n)} \to 0\}}$$

How does delay impact capacity?



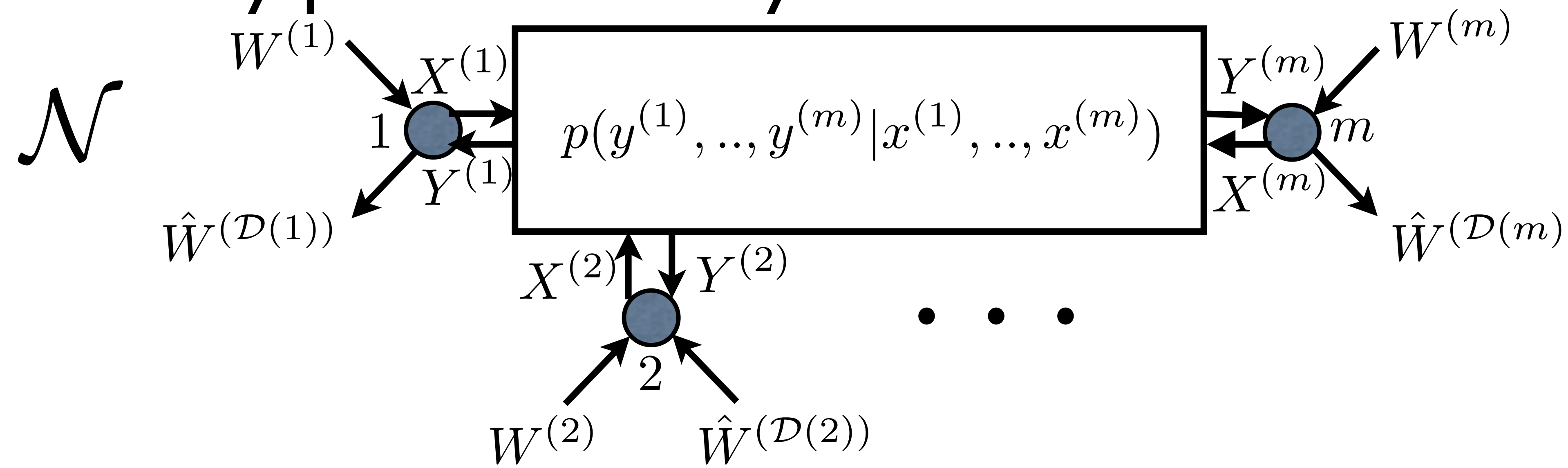
Some literature:

Gaussian channels with delayed feedback [Yanagi, 1995]
Relay channel with delay [van der Muelen & Vanroose, 2007]
Relay networks with delay [El Gamal, Hassanpour, Mammen 2007]
Cut-set bounds for generalized networks with positive delay [Fong & Yeung 2012]
On network coding for acyclic networks with delays [Prasad & Rajan 2012]

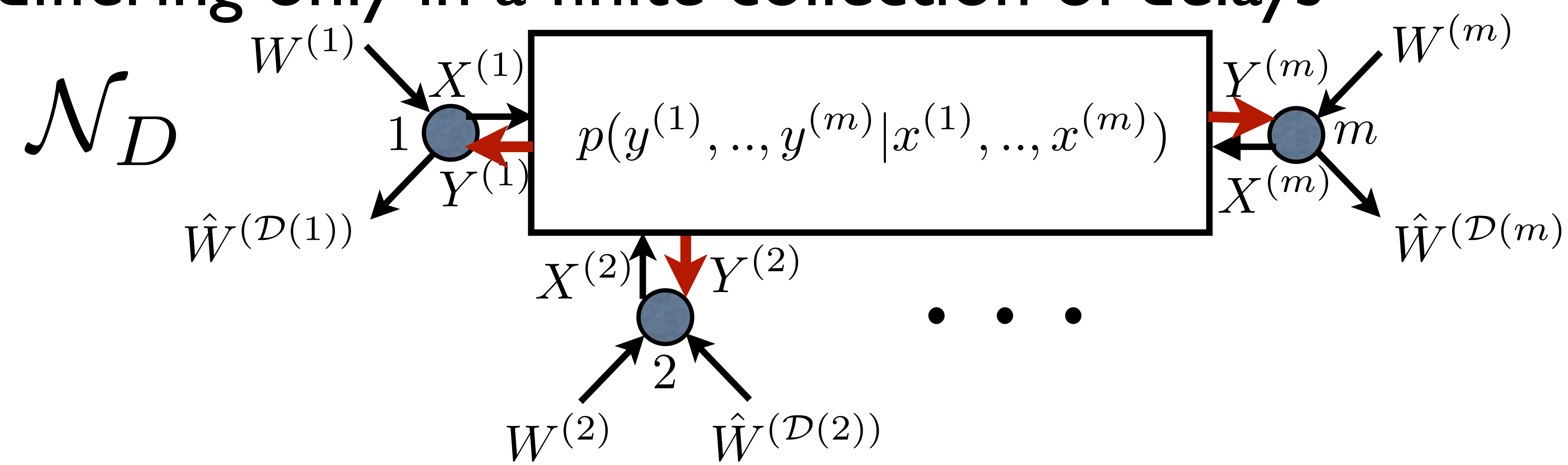
Theorem [Effros, 2012]:

Delay has no impact on network capacity.

Given any pair of memoryless networks

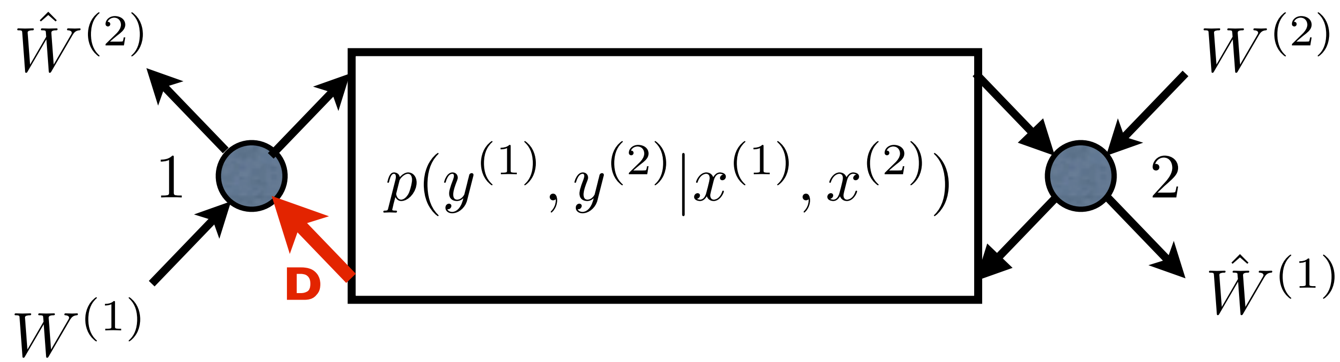


differing only in a finite collection of delays



$$\text{Capacity}(\mathcal{N}) = \text{Capacity}(\mathcal{N}_D)$$

Proof: $\text{Capacity}(\mathcal{N}) \supseteq \text{Capacity}(\mathcal{N}_D)$: A code for $\mathcal{N}_D$...



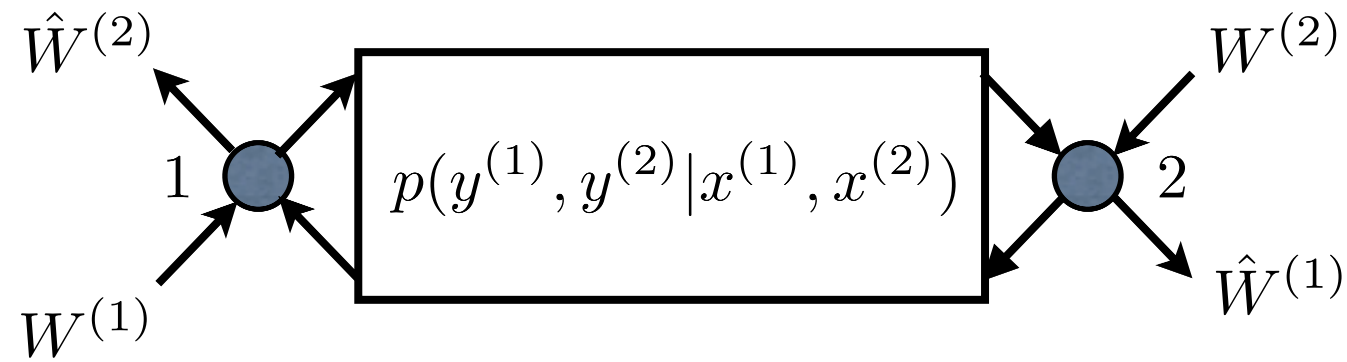
t	$X^{(1)}$	$X^{(2)}$	$Y^{(1)}$	$Y^{(2)}$
1	$X_1^{(1)}(W^{(1)})$	$X_1^{(2)}(W^{(2)})$	$-$	$Y_1^{(2)}$
2	$X_2^{(1)}(W^{(1)})$	$X_2^{(2)}(W^{(2)}, Y_1^{(2)})$	$Y_1^{(1)}$	$Y_2^{(2)}$
3	$X_3^{(1)}(W^{(1)}, Y_1^{(1)})$	$X_3^{(2)}(Y^{(2)}, Y_{1:2}^{(2)})$	$Y_2^{(1)}$	$Y_3^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$X_n^{(1)}(W^{(1)}, Y_{1:n-2}^{(1)})$	$X_n^{(2)}(W^{(2)}, Y_{1:n-1}^{(2)})$	$Y_{n-1}^{(1)}$	$Y_n^{(2)}$
	$\hat{W}^{(2)}(W^{(1)}, Y_{1:n-1}^{(1)})$	$\hat{W}^{(1)}(W^{(2)}, Y_{1:n}^{(2)})$		

Rate: $(R^{(1)}, R^{(2)})$

Error Prob: $P_e^{(n)}$

Proof: $\text{Capacity}(\mathcal{N}) \supseteq \text{Capacity}(\mathcal{N}_D)$:

$\ldots$ can run on $\mathcal{N}$ without any performance loss.



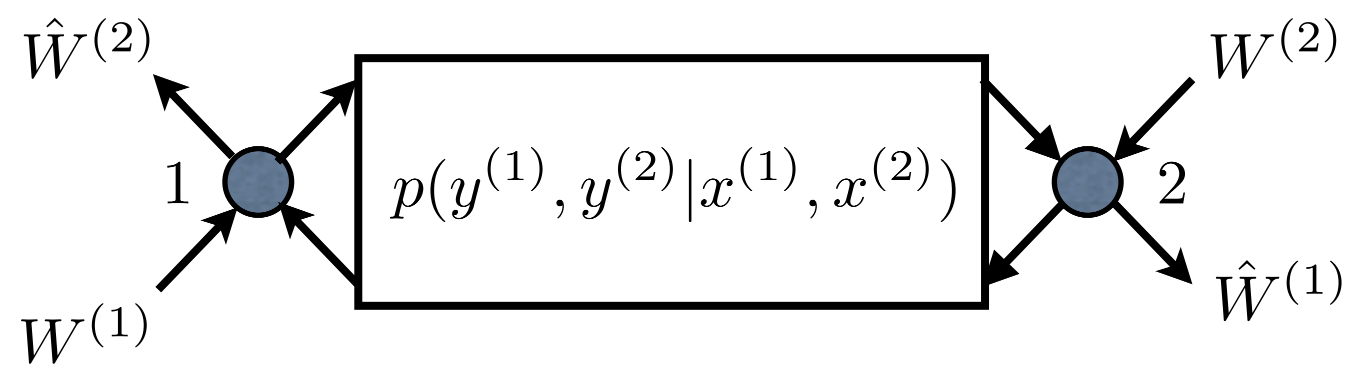
t	$X^{(1)}$	$X^{(2)}$	$Y^{(1)}$	$Y^{(2)}$
1	$X_1^{(1)}(W^{(1)})$	$X_1^{(2)}(W^{(2)})$	$Y_1^{(1)}$	$Y_1^{(2)}$
2	$X_2^{(1)}(W^{(1)})$	$X_2^{(2)}(W^{(2)}, Y_1^{(2)})$	$Y_2^{(1)}$	$Y_2^{(2)}$
3	$X_3^{(1)}(W^{(1)}, Y_1^{(1)})$	$X_3^{(2)}(Y^{(2)}, Y_{1:2}^{(2)})$	$Y_3^{(1)}$	$Y_3^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$X_n^{(1)}(W^{(1)}, Y_{1:n-2}^{(1)})$	$X_n^{(2)}(W^{(2)}, Y_{1:n-1}^{(2)})$	$Y_n^{(1)}$	$Y_n^{(2)}$
	$\hat{W}^{(2)}(W^{(1)}, Y_{1:n-1}^{(1)})$	$\hat{W}^{(1)}(W^{(2)}, Y_{1:n}^{(2)})$		

Rate: $(R^{(1)}, R^{(2)})$

Error Prob: $P_e^{(n)}$

Proof: $\text{Capacity}(\mathcal{N}) \subseteq \text{Capacity}(\mathcal{N}_D)$:

A code designed for no delay...



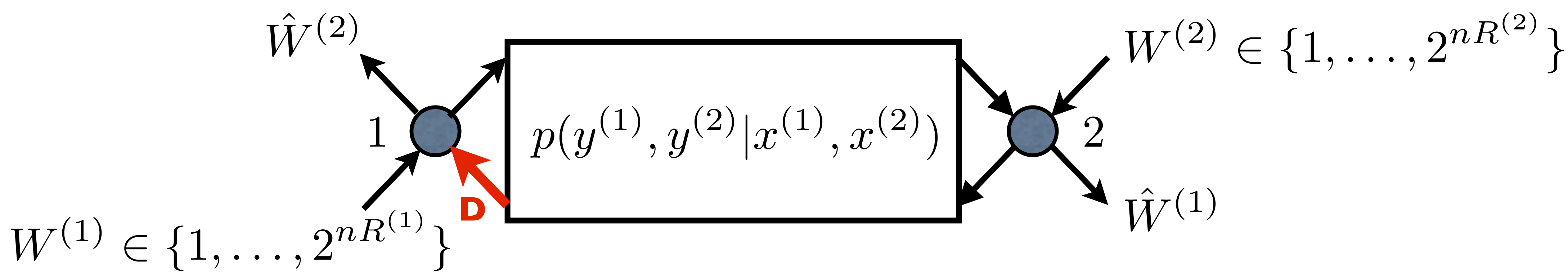
t	$X^{(1)}$	$X^{(2)}$	$Y^{(1)}$	$Y^{(2)}$
1	$X_1^{(1)}(W^{(1)})$	$X_1^{(2)}(W^{(2)})$	$Y_1^{(1)}$	$Y_1^{(2)}$
2	$X_2^{(1)}(W^{(1)}, Y_1^{(1)})$	$X_2^{(2)}(W^{(2)}, Y_1^{(2)})$	$Y_2^{(1)}$	$Y_2^{(2)}$
3	$X_3^{(1)}(W^{(1)}, Y_{1:2}^{(1)})$	$X_3^{(2)}(W^{(2)}, Y_{1:2}^{(2)})$	$Y_3^{(1)}$	$Y_3^{(2)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	$X_n^{(1)}(W^{(1)}, Y_{1:n-1}^{(1)})$	$X_n^{(1)}(W^{(2)}, Y_{1:n-1}^{(2)})$	$Y_n^{(1)}$	$Y_n^{(2)}$
	$\hat{W}^{(2)}(W^{(1)}, Y_{1:n}^{(1)})$	$\hat{W}^{(1)}(W^{(2)}, Y_{1:n}^{(2)})$		

Rate: $(R^{(1)}, R^{(2)})$

Error Prob: $P_e^{(n)}$

Proof: $\text{Capacity}(\mathcal{N}) \subseteq \text{Capacity}(\mathcal{N}_D)$:

... can be run with delay, but the cost seems to be high.

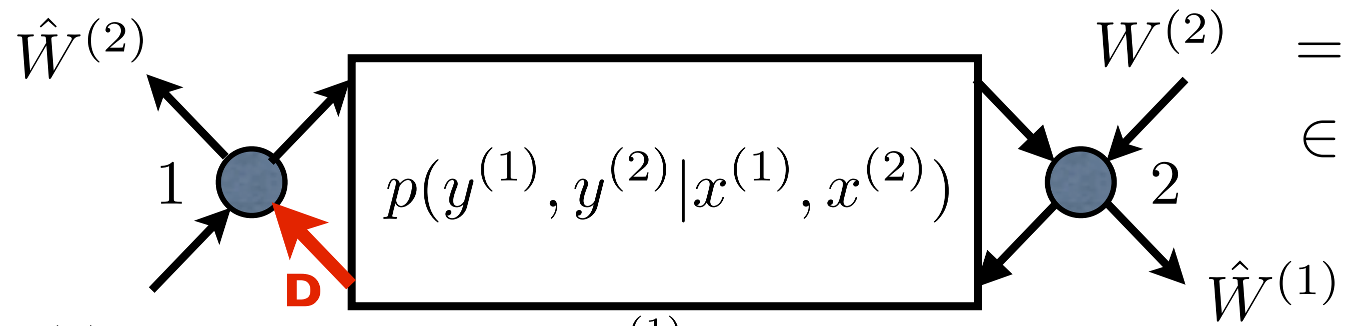


t	$X^{(1)}$	$X^{(2)}$	$Y^{(1)}$	$Y^{(2)}$
1	$X_1^{(1)}(W^{(1)})$	$X_1^{(2)}(W^{(2)})$	–	$Y_1^{(2)}$
2	–	–	$Y_1^{(1)}$	–
3	$X_2^{(1)}(W^{(1)}, Y_1^{(1)})$	$X_2^{(2)}(W^{(2)}, Y_1^{(2)})$	–	$Y_2^{(2)}$
2	–	–	$Y_2^{(1)}$	–
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2n-1$	$X_n^{(1)}(W^{(1)}, Y_{1:n-1}^{(1)})$	$X_n^{(2)}(W^{(2)}, Y_{1:n-1}^{(2)})$	–	$Y_n^{(2)}$
$2n$	–	–	$Y_n^{(1)}$	–
	$\hat{W}^{(2)}(W^{(1)}, Y_{1:n}^{(1)})$	$\hat{W}^{(1)}(W^{(2)}, Y_{1:n}^{(2)})$		

Rate: $(R^{(1)}/2, R^{(2)}/2)$

Error Prob: $P_e^{(n)}$

Proof: $\text{Capacity}(\mathcal{N}) \subseteq \text{Capacity}(\mathcal{N}_D)$: Better machinery reduces the cost.

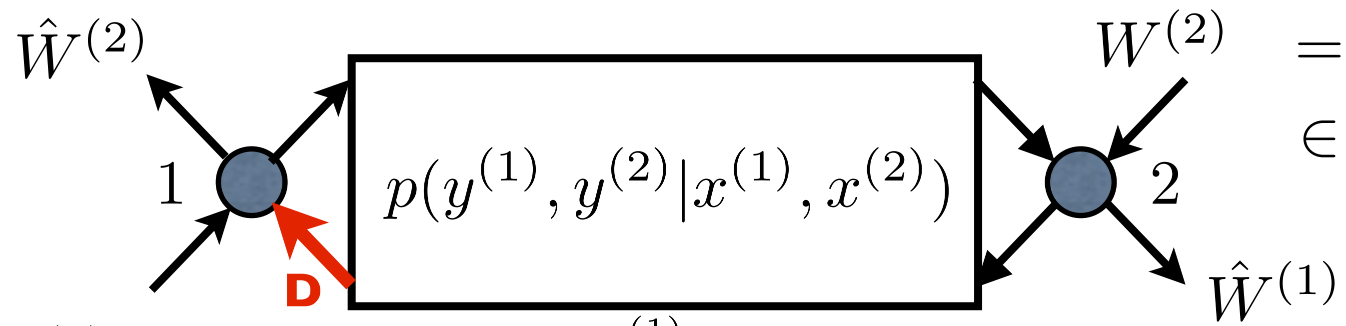


$$W^{(2)} = (W^{(2)}(1), W^{(2)}(2)) \in \{1, \ldots, 2^{2nR^{(2)}}\}$$

$$W^{(1)} = (W^{(1)}(1), W^{(1)}(2)) \in \{1, \ldots, 2^{2nR^{(1)}}\}$$

t	$X^{(1)}$	$X^{(2)}$	$Y^{(1)}$	$Y^{(2)}$
1	$X_1^{(1)}(W^{(1)}(1))$	$X_1^{(2)}(W^{(2)}(1))$		$Y_1^{(2)}(1)$
2			$Y_1^{(1)}(1)$	
3	$X_2^{(1)}(W^{(1)}(1), Y_1^{(1)}(1))$	$X_2^{(2)}(W^{(2)}(1), Y_1^{(2)}(1))$		$Y_2^{(2)}(1)$
4			$Y_2^{(1)}(1)$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2n$			$Y_n^{(1)}(1)$	

Proof: $\text{Capacity}(\mathcal{N}) \subseteq \text{Capacity}(\mathcal{N}_D)$:
Better machinery reduces the cost to zero.



$$W^{(2)} = (W^{(2)}(1), W^{(2)}(2)) \in \{1, \ldots, 2^{2nR^{(2)}}\}$$

$$W^{(1)} = (W^{(1)}(1), W^{(1)}(2)) \in \{1, \ldots, 2^{2nR^{(1)}}\}$$

t	$X^{(1)}$	$X^{(2)}$	$Y^{(1)}$	$Y^{(2)}$
1	$X^{(1)}_1(W^{(1)}(1))$	$X^{(2)}_1(W^{(2)}(1))$		$Y^{(2)}_1(1)$
2	$X^{(1)}_1(W^{(1)}(2))$	$X^{(2)}_1(W^{(2)}(2))$	$Y^{(1)}_1(1)$	$Y^{(2)}_1(2)$
3	$X^{(1)}_2(W^{(1)}(1), Y^{(1)}_1(1))$	$X^{(2)}_2(W^{(2)}(1), Y^{(2)}_1(1))$	$Y^{(1)}_1(2)$	$Y^{(2)}_2(1)$
4	$X^{(1)}_2(W^{(1)}(2), Y^{(1)}_1(2))$	$X^{(2)}_2(W^{(2)}(2), Y^{(2)}_1(2))$	$Y^{(1)}_2(1)$	$Y^{(2)}_2(2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2n$	$X^{(1)}_n(W^{(1)}(2), Y^{(1)}_{1:n-1}(2))$	$X^{(2)}_n(W^{(2)}(2), Y^{(2)}_{1:n-1}(2))$	$Y^{(1)}_n(1)$	$Y^{(2)}_n(2)$
$2n+1$			$Y^{(1)}_n(2)$	

$$\hat{W}^{(2)} = (\hat{W}^{(2)}(W^{(1)}(1), Y^{(1)}_{1:n}(1)), \hat{W}^{(2)}(W^{(1)}(2), Y^{(1)}_{1:n}(2)))$$

$$\hat{W}^{(1)} = (\hat{W}^{(1)}(W^{(2)}(1), Y^{(2)}_{1:n}(1)), \hat{W}^{(1)}(W^{(2)}(2), Y^{(2)}_{1:n}(2)))$$

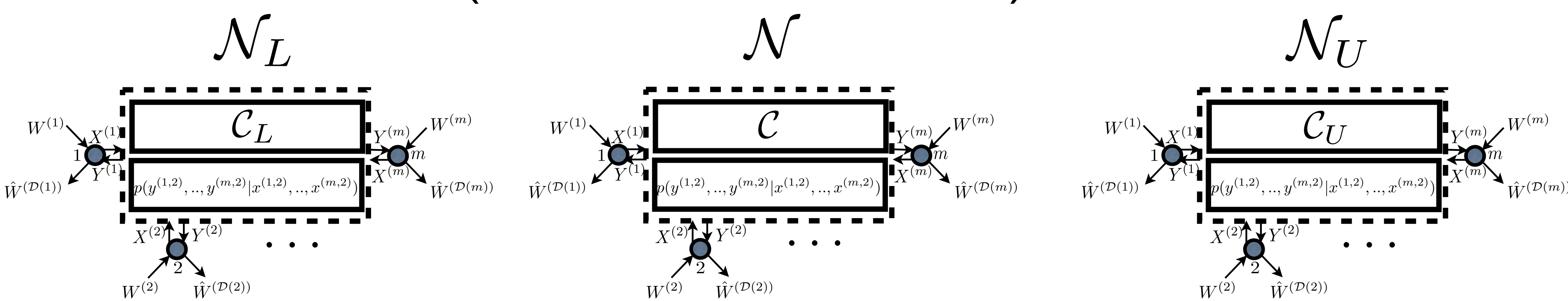
$$\text{Rate: } \left(R^{(1)}\tfrac{2n}{2n+1}, R^{(2)}\tfrac{2n}{2n+1}\right), \quad \text{Error Prob: } 2P_e^{(n)}$$

Is there a path to a scalable information theory?

[Koetter, Effros, Medard, 2009]

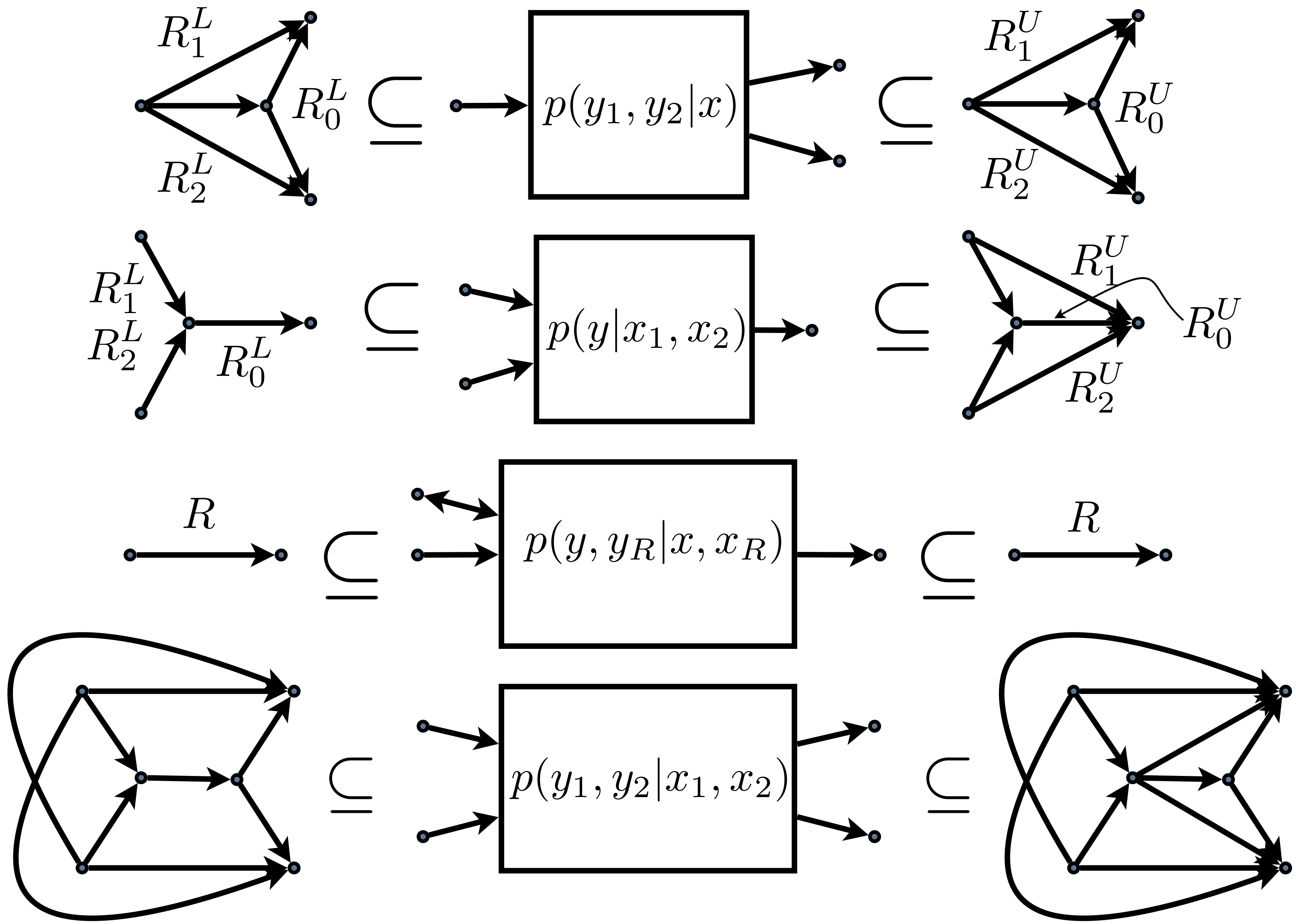
Derive bounding models that compose.

$(\mathcal{C}_L, \mathcal{C}_U)$ are (lower, upper) bounding models for $\mathcal{C}$ (written $\mathcal{C}_L \subseteq \mathcal{C} \subseteq \mathcal{C}_U$) iff



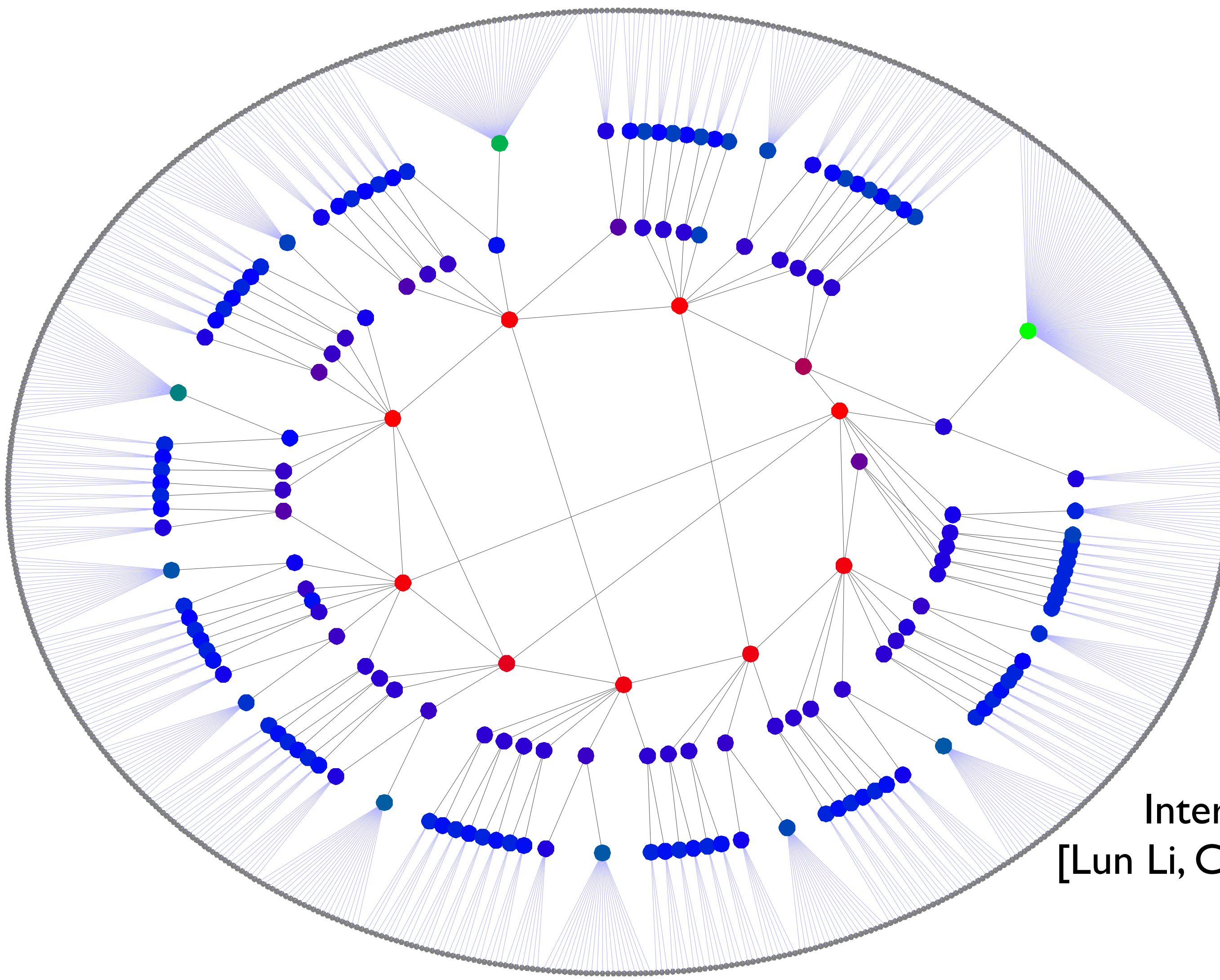
$$\text{Capacity}(\mathcal{N}_L) \subseteq \text{Capacity}(\mathcal{N}) \subseteq \text{Capacity}(\mathcal{N}_U)$$

Our models are made of lossless links.



[Koetter, Effros, & Medard 2009][Wong & Effros 2012]

We can bound the network capacity by bounding the network coding capacity.

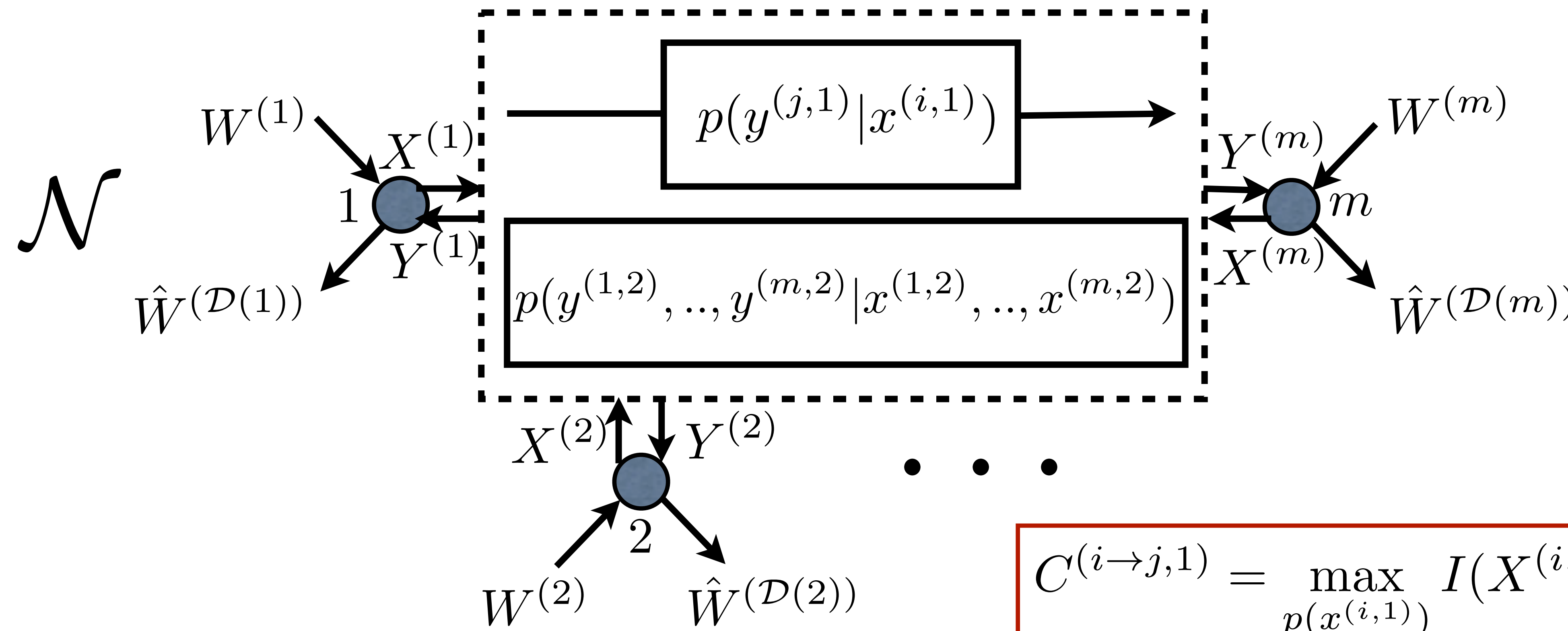


Internet topology
[Lun Li, Caltech Ph.D. 2007]

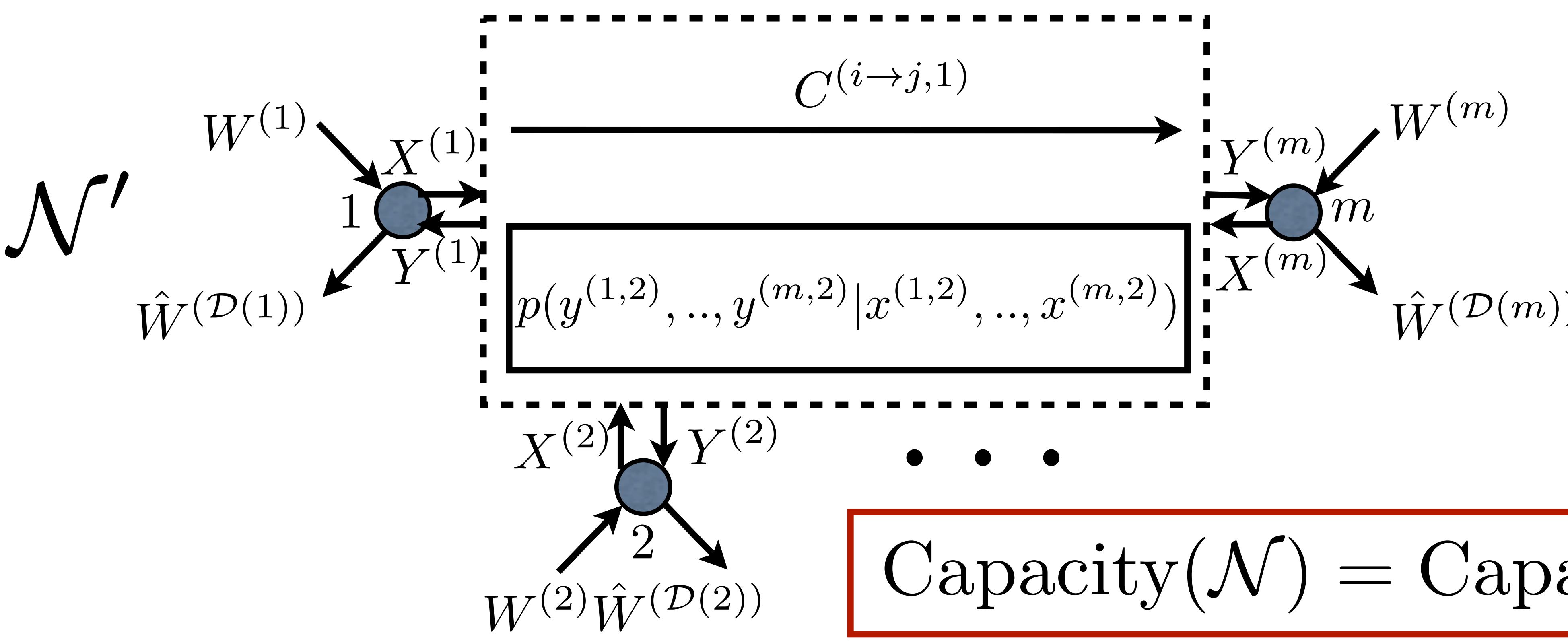
There exist computational tools for bounding network coding capacities.

(e.g., [Yeung, 1997][Subramanian et al., 2008])

Example: A noisy channels is bounded (above and below) by a lossless link of the same capacity.



$$C^{(i\to j,1)} = \max_{p(x^{(i,1)})} I(X^{(i,1)};Y^{(j,1)})$$

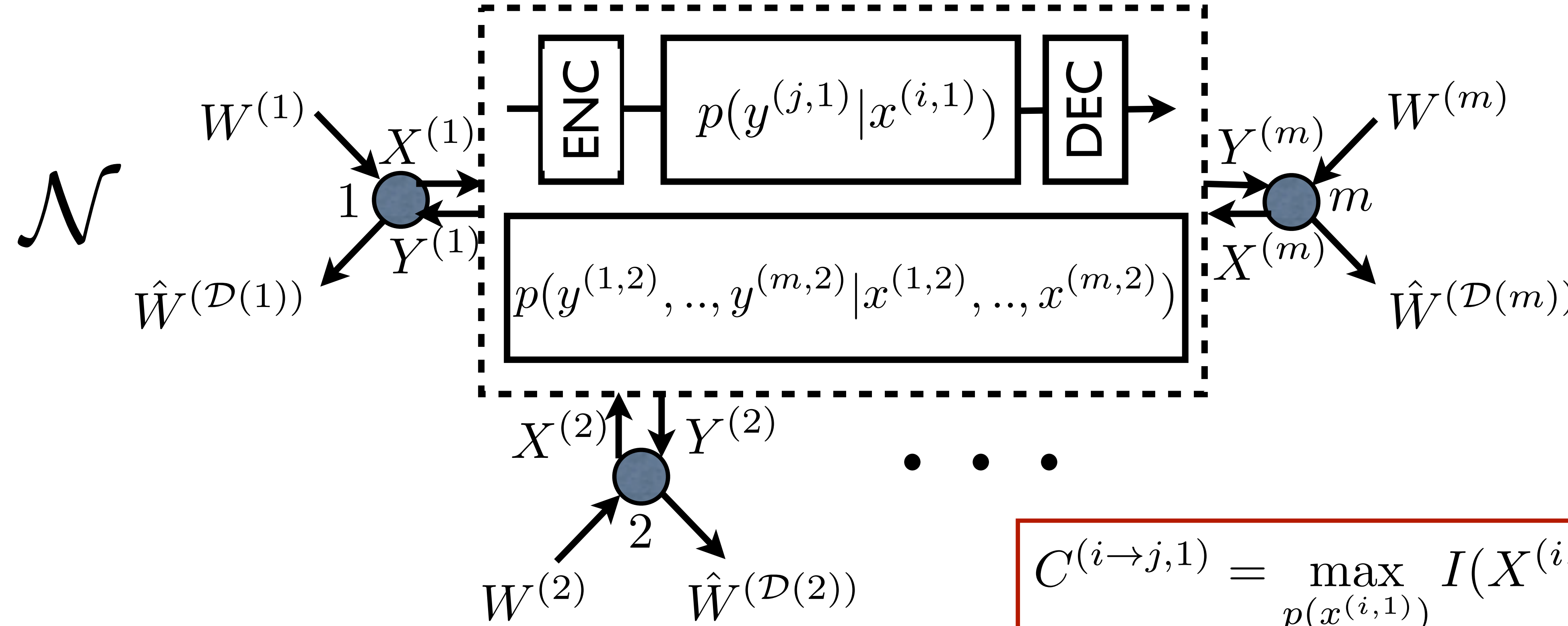


$$\text{Capacity}(\mathcal{N}) = \text{Capacity}(\mathcal{N}')$$

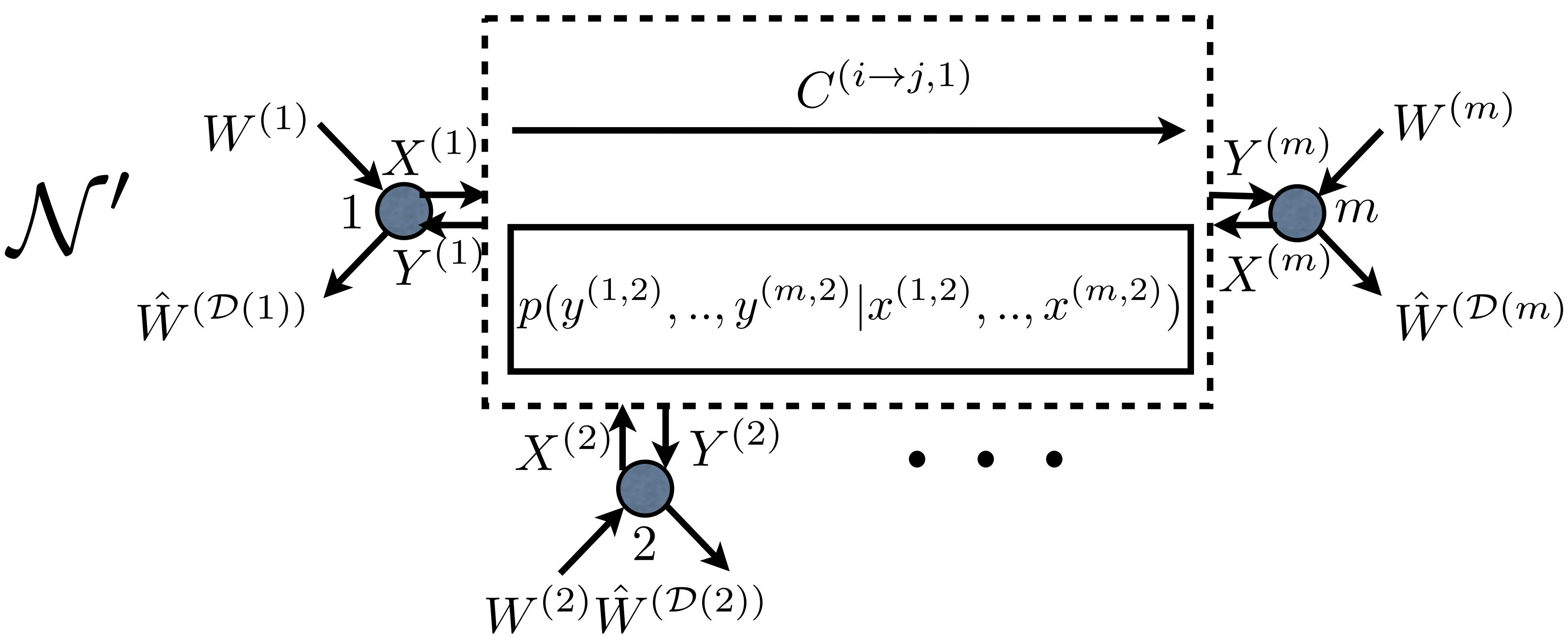
Michelle Effros, LCCC 2012

Proof: $\text{Capacity}(\mathcal{N}) \supseteq \text{Capacity}(\mathcal{N}') :$

A channel code makes the noisy channel act like a link.

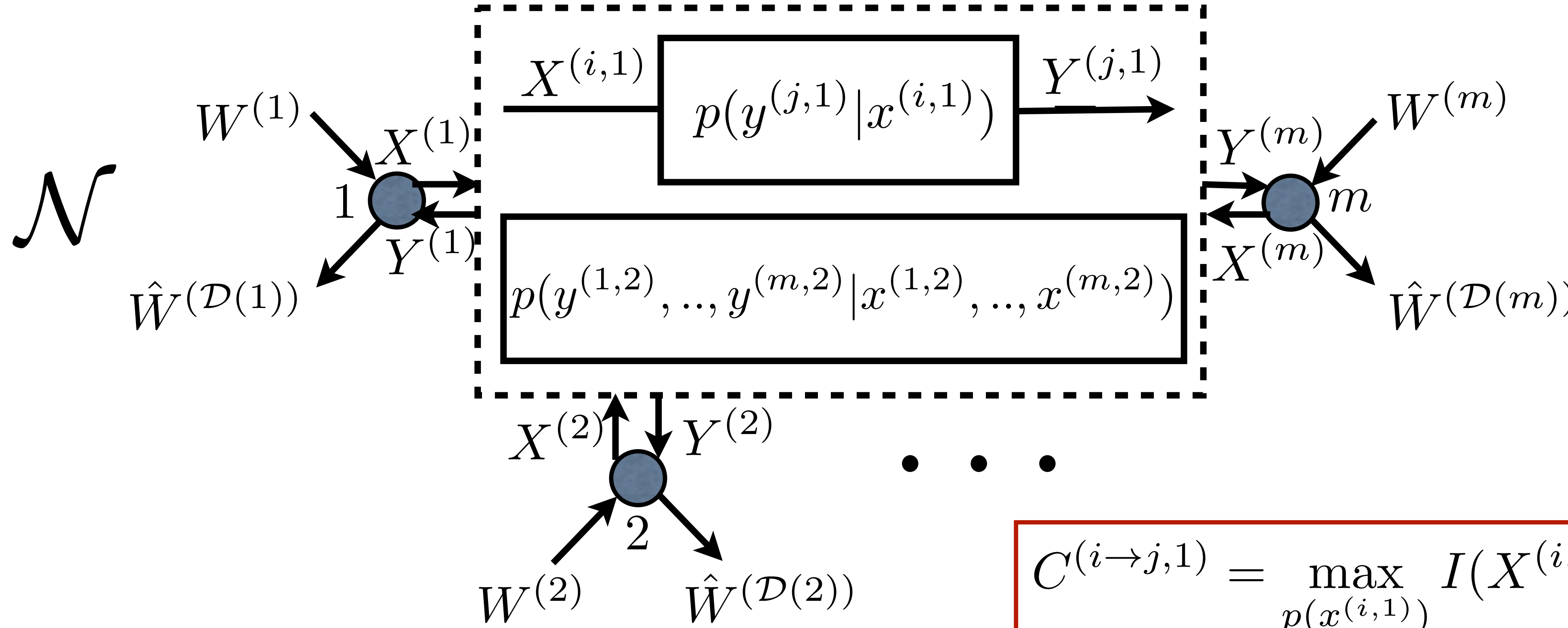


$$C^{(i \to j,1)} = \max_{p(x^{(i,1)})} I(X^{(i,1)}; Y^{(j,1)})$$

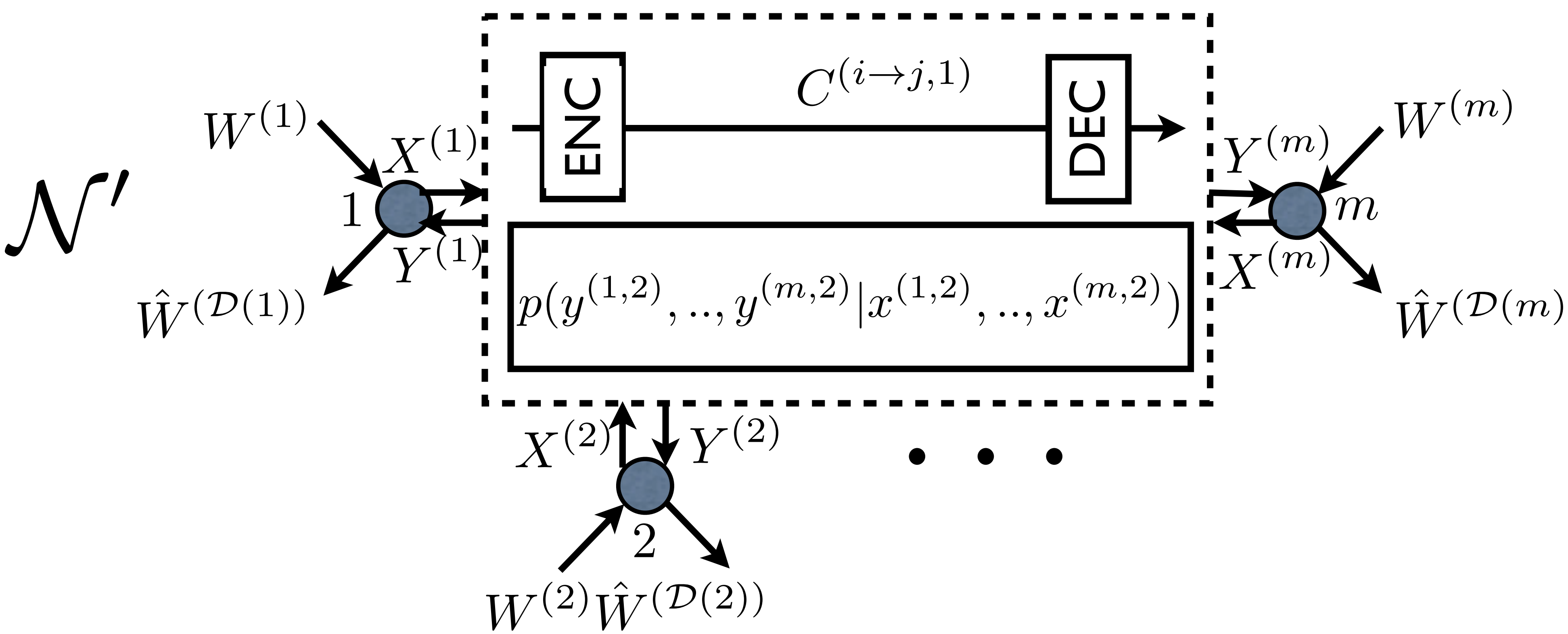


Proof: $\text{Capacity}(\mathcal{N}) \subseteq \text{Capacity}(\mathcal{N}')$:

A source code makes the link act like a noisy channel.



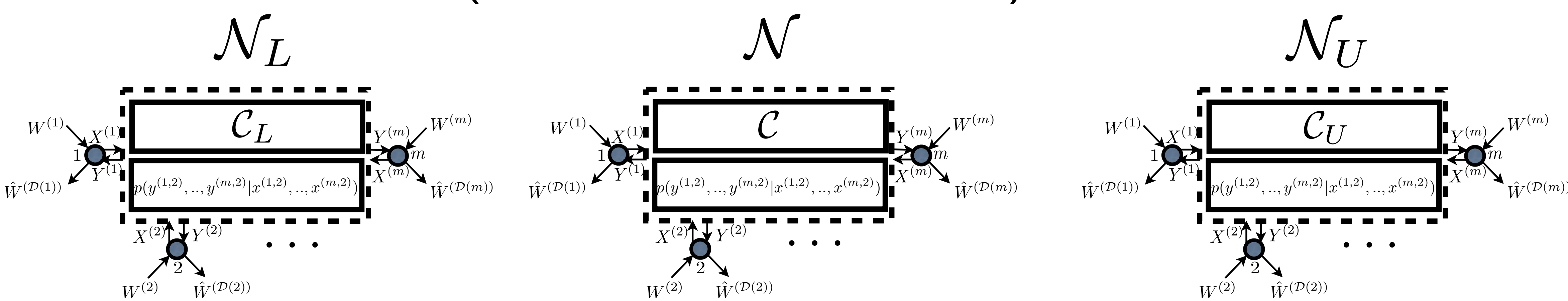
$$C^{(i\to j,1)} = \max_{p(x^{(i,1)})} I(X^{(i,1)};Y^{(j,1)})$$



Can we move beyond capacity to controls?

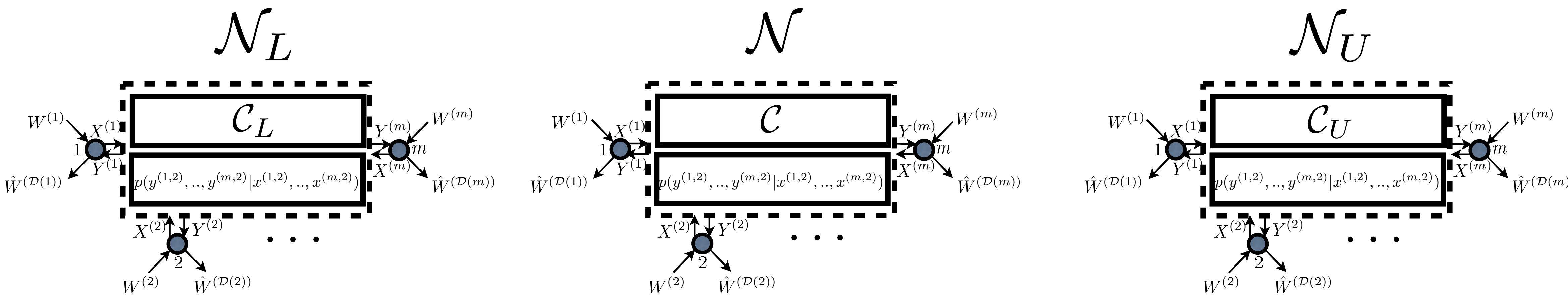
$$\begin{aligned} C &= \{\text{controls objectives individually achievable across } \mathcal{N}\} \\ \text{Controls}(\mathcal{N}) &= \{c^k \in C^* : (c_1, \ldots, c_k) \text{ simultaneously achievable across } \mathcal{N}\} \end{aligned}$$

$(\mathcal{C}_L, \mathcal{C}_U)$ are (lower, upper) bounding models for $\mathcal{C}$ (written $\mathcal{C}_L \subseteq \mathcal{C} \subseteq \mathcal{C}_U$) iff



$$\text{Controls}(\mathcal{N}_L) \subseteq \text{Controls}(\mathcal{N}) \subseteq \text{Controls}(\mathcal{N}_U)$$

Can we move beyond capacity to controls?

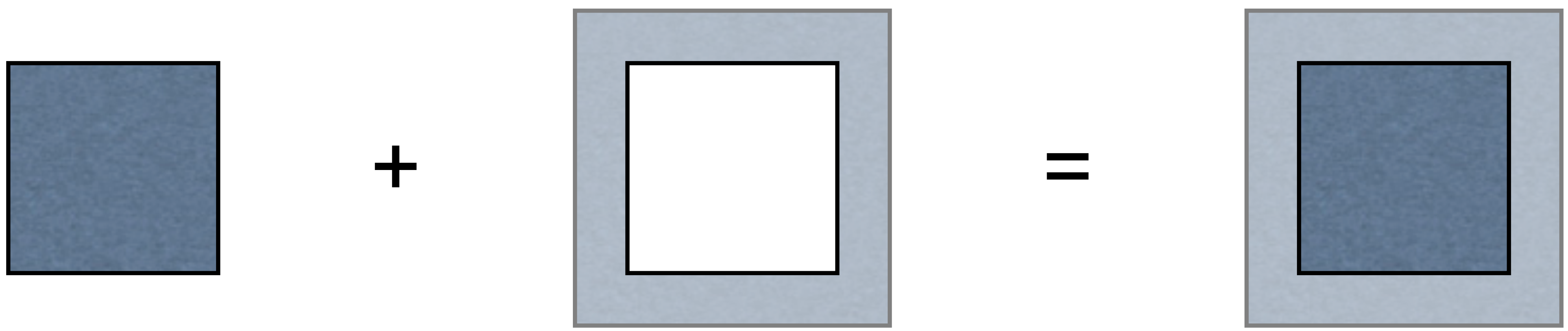


$$\text{Controls}(\mathcal{N}_L) \subseteq \text{Controls}(\mathcal{N}) \subseteq \text{Controls}(\mathcal{N}_U)$$

Design for the lower bounding network.
If the lower bound cannot meet the desired objectives,
test for achievability on the upper bounding model.

How should we measure communications performance for controls?

Given: Networks A & B



controls solution across network A

a bit of machinery

controls solution across network B

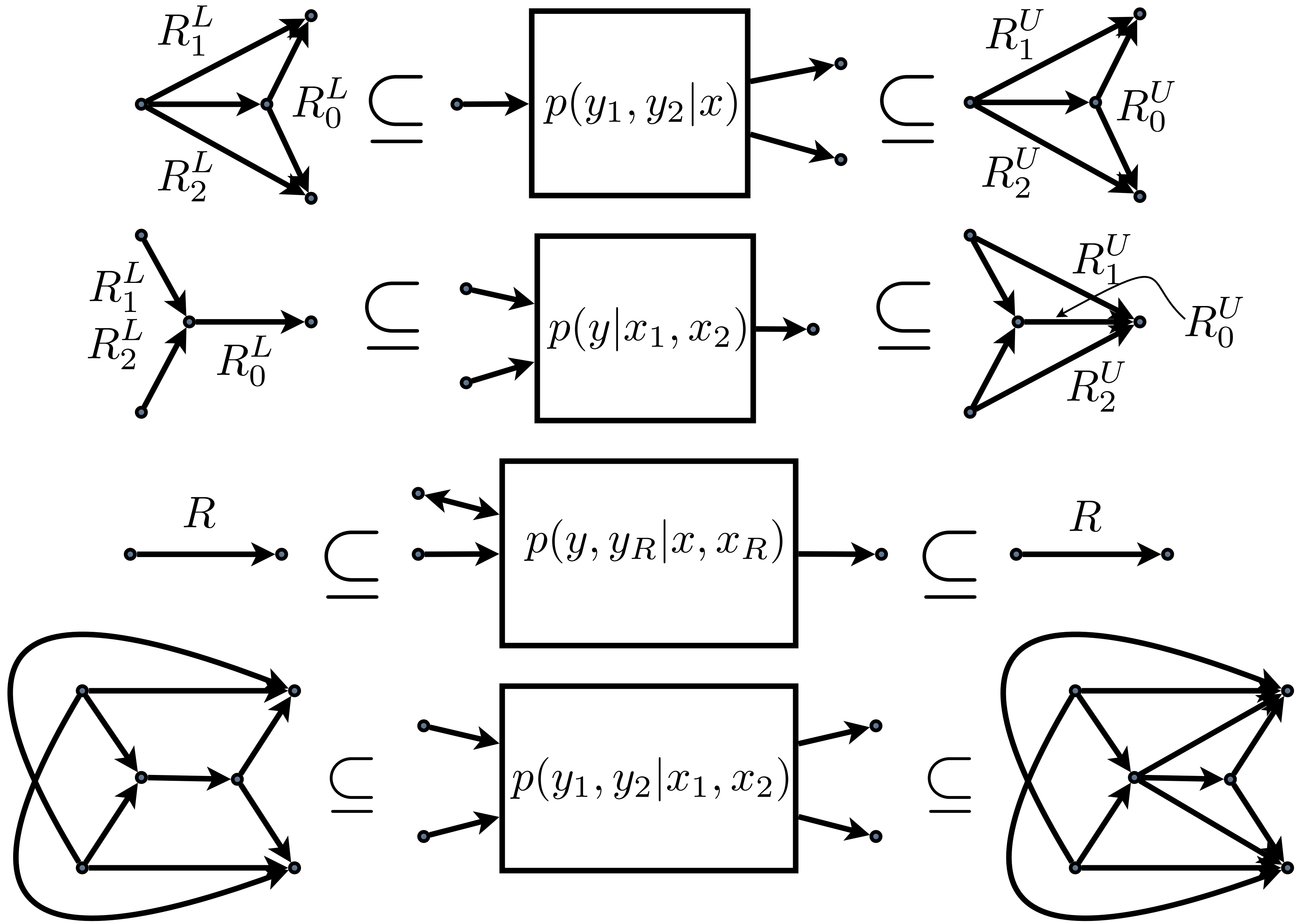
What must the machinery promise?

The literature suggests many performance measures.

Quantization noise [Elia & Mitter 2001, Xiao, et al. 2005]
Delay constraints [Berry & Gallager 2002]
Packet arrival probabilities [Sinopoli et al. 2005, Imer et al. 2006]
Data rates / quantization [Tatikonda & Mitter 2005][Nair et al. 2007]
Estimation error [Tatikonda & Mitter 2005]
Anytime capacity [Sahai & Mitter 2006]
Maximal admissible delay [Fan & Arcak, 2006]
Minimal average delay [Bettesh & Shamai 2006]

AND MANY MORE...

Prior models generalize under some of these measures.
Ex: Anytime reliability (with parameter change)



How do the models change with the measure?
Can we find models for all measures?

Reduction provides a path towards developing a computational information theory.

This tool was originally explored for capacity but has since been generalized to other problems:

Joint source-channel coding [Jalali & Effros 2010, 2011]

Non-ergodic channels [Bakshi, Effros & Ho 2011]

Secure capacity [Dikaliotis, Yao, Ho, Effros & Kliewer 2012]

Noiseless components [Ho, Effros, & Jalali 2010]

...

The same tool may provide a useful tool for simplifying the interaction between communications and controls.