Scheduling for Compute & Forward Networks

DAVID RAMIREZ
BEHNAAM AAZHANG

Challenging Orthogonality

CDMA

Spread out your signal to share your resources

Network Coding

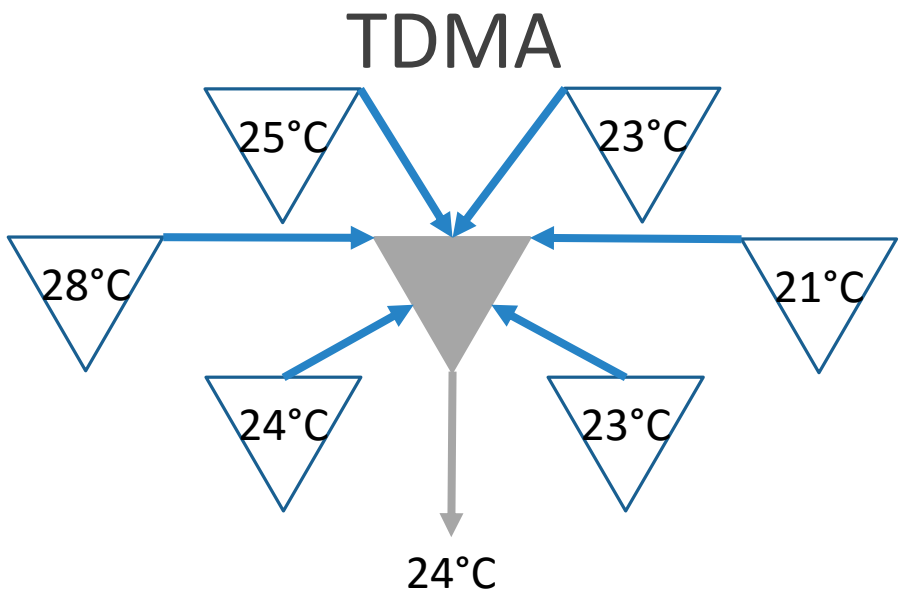
Obtain combinations of signals to maximize flow

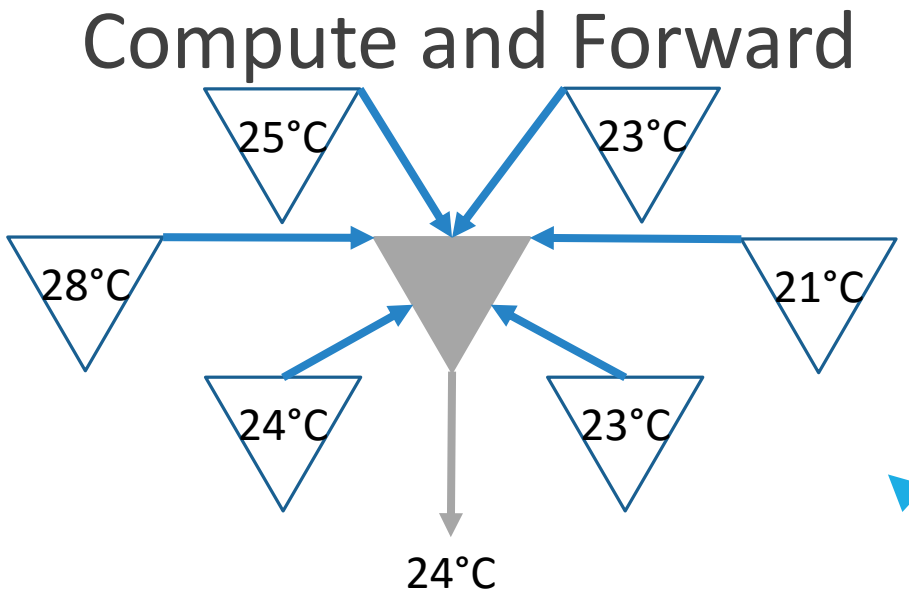
Compute and Forward

Compute a function of the superimposed signals

Orthogonality: Needed or Wanted?

Orthogonality requires separate time block per user





Channel *computes* for us the superimposed signals

$$a_1 25 + a_2 23 + a_3 21 + a_4 23 + a_5 24 + a_6 28$$

Nazer, Bobak, and Michael Gastpar. "Compute-and-forward: Harnessing interference through structured codes." *IEEE Transactions on Information Theory, 2011*

Nokleby, Matthew, et al. "Relays that cooperate to compute." *IEEE International Symposium on Wireless Communication Systems (ISWCS)*, 2012.

Compute and Forward

Channel *computes* for us, but we need a structured code

- Lattice codes useful for superposition

N nodes with transmit power P, function coefficients $\mathbf{a}$, slow fading channels and channel gains $\mathbf{h}$

TDMA

$$r = \frac{1}{N}\log\left(1+\frac{P\|h_i\|^2}{1}\right)$$

Compute and Forward

$$r = \log^+\left(\frac{1+P\|\mathbf{h}\|^2}{\|\mathbf{a}\|^2 + P\|\mathbf{h}\|^2\|\mathbf{a}\|^2 - P|\mathbf{h}^*\mathbf{a}|^2}\right)$$

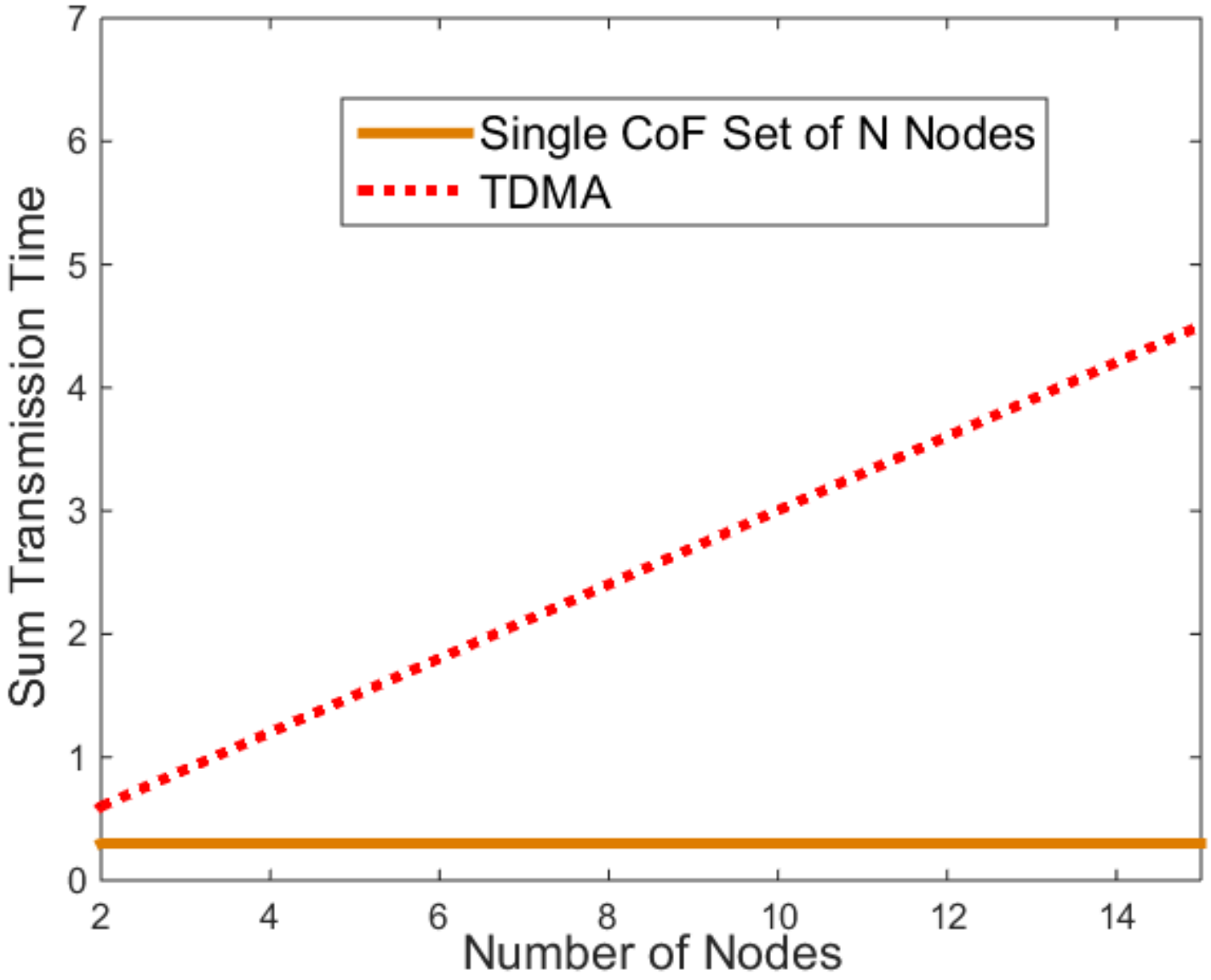
Interested in minimizing time to transmit L bits, i.e. $t = \frac{L}{r}$

Nazer, Bobak, and Michael Gastpar. "Compute-and-forward: Harnessing interference through structured codes." *IEEE Transactions on Information Theory, 2011*
Erez, Uri, and Ram Zamir. "Achieving 1/2 log (1+ SNR) on the AWGN channel with lattice encoding and decoding." *IEEE Transactions on Information Theory, 2004*

Single Ideal Set

- P=20dB
- h=a=1
- L=1

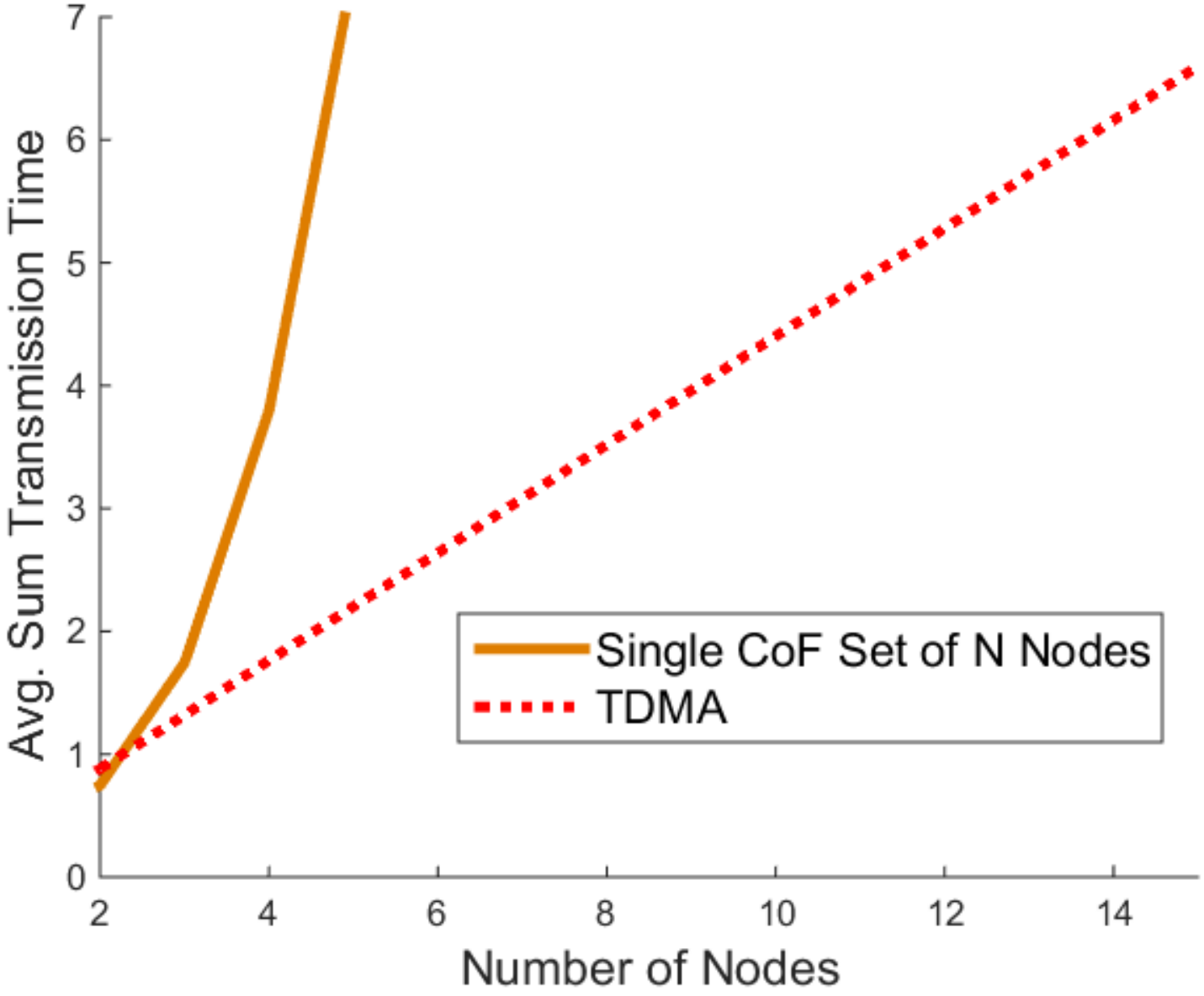
We don't live in an ideal world



Single Set

- P=20dB
- h=0.3+C$\mathcal{N}$(0,1)
- a_i=1
- L=1

Rate is low due to mismatch between channels and coefficients

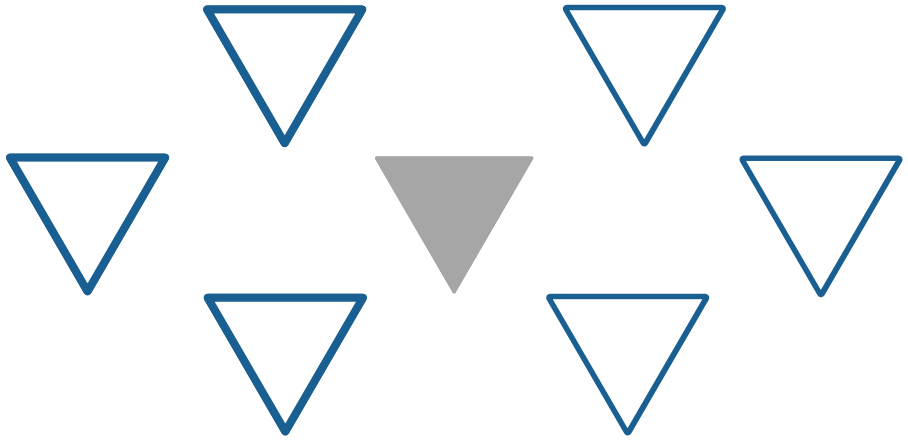


Problem Formulation

- Minimize the time to empty a given queue

$$\min_{\mathbf{s}} \sum_{s \in \mathbf{s}} t_s$$

$$s.t. \quad \sum_{s \in \mathbf{s}} r_s t_s \mathbf{1}_s(i) = L, \quad \forall i = 1, \ldots, N$$



Problem Formulation

$$\min_{\mathbf{s}} \sum_{S \in \mathbf{s}} t_S$$

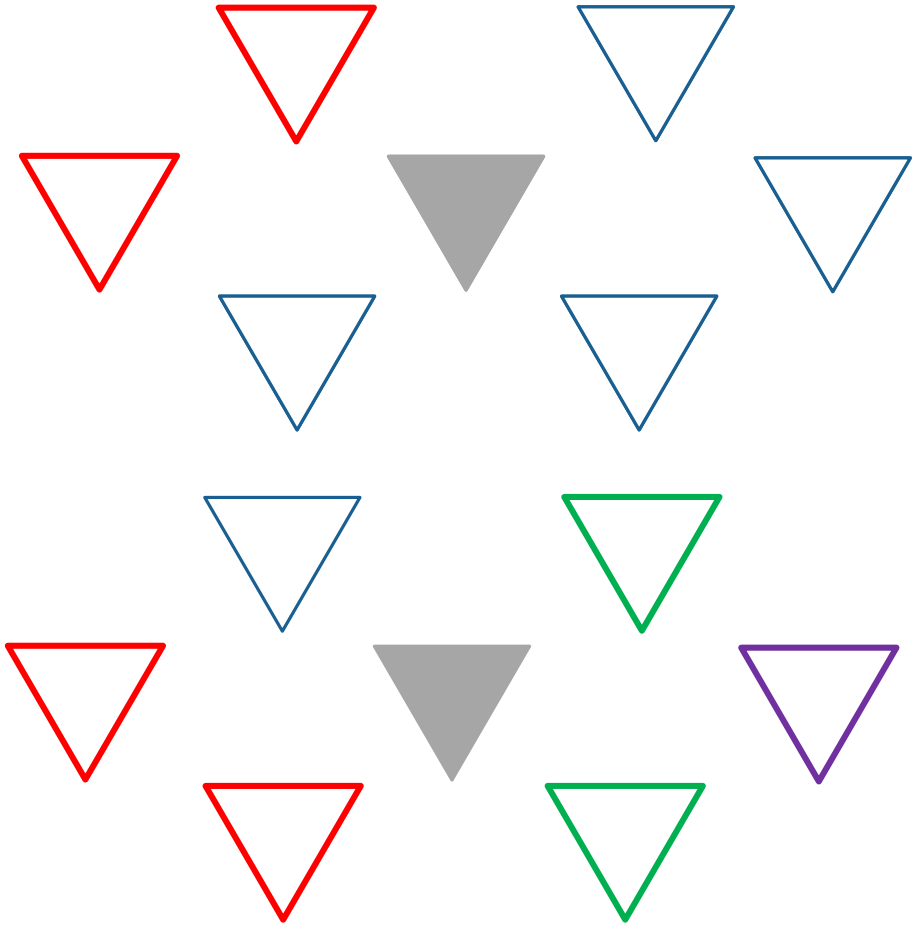
$$s.t. \quad \sum_{S \in \mathbf{s}} r_S t_S \mathbf{1}_S(i) = L, \quad \forall i = 1,\ldots,N$$

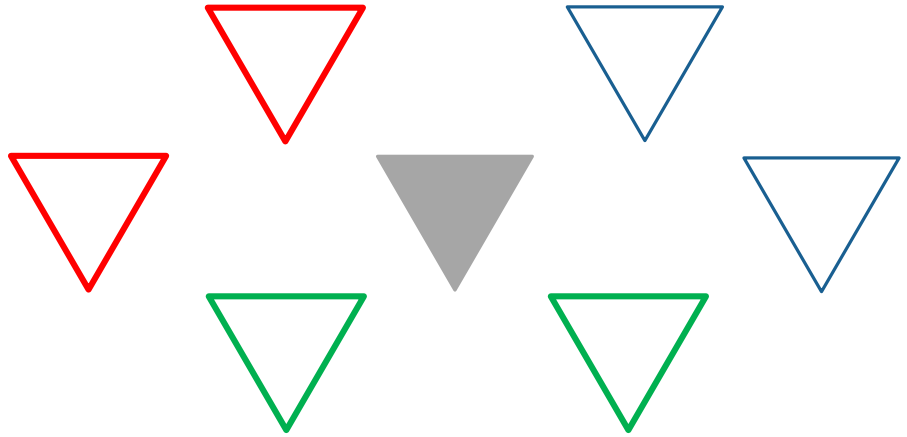
Non-monotonic rate function $r = 0 \quad if \quad \|\mathbf{a}\|^2 \geq 1 + P\|\mathbf{h}\|^2$

NP-Hard

- NP from recognition version: are there N sets that can be activated to transmit L bits?

How many possible partitions exist?





$$\sum_{k=1}^{N}\left(\frac{1}{k!}\sum_{i=0}^{k}(-1)^{i}\binom{k}{i}(k-i)^{N}\right)$$

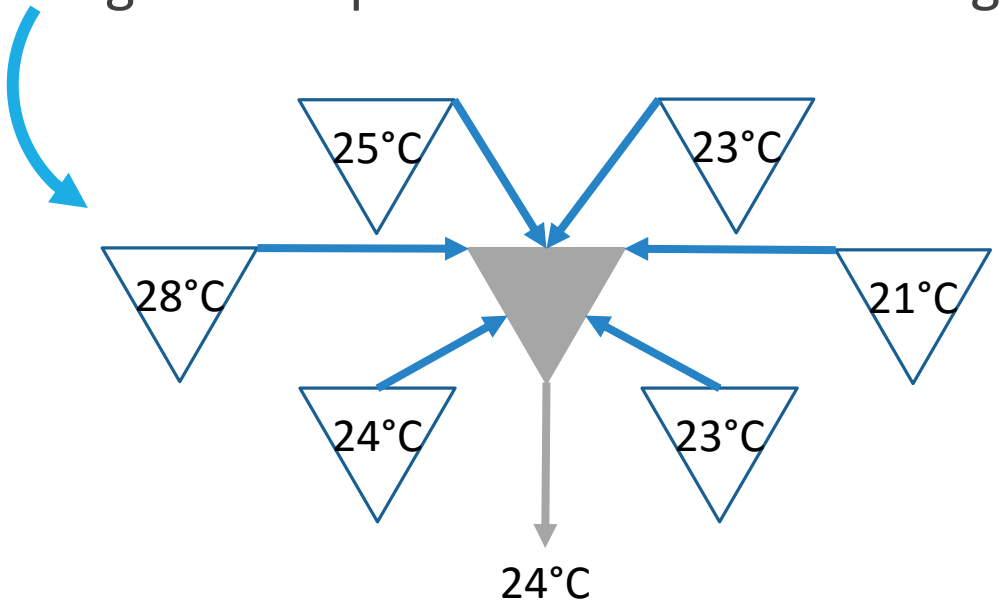
Numbers for Perspective

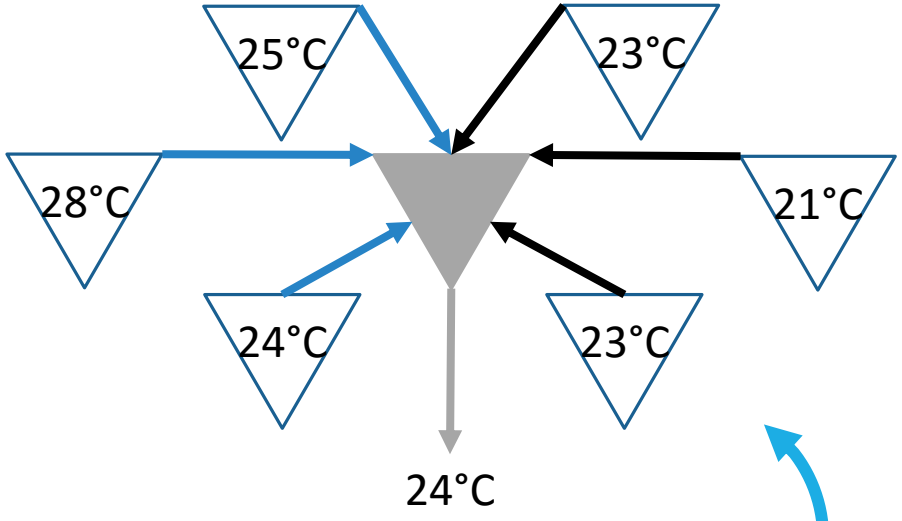
		Number of combinations to find…
N	**2^N**	**Globally Optimal**
1	2	1
2	4	2
3	8	5
4	16	15
5	32	52
6	64	203
7	128	877
8	256	4,140
9	512	21,147
10	1,024	115,975
15	32,768	1.383 e+09

- 2^N is your typical combinatorial problem
- Faster than exponential with N is *scary*

Grouping Intuition with CoF

Higher computation rate if channel gains similar among themselves





More orthogonal partitions means more transmission rounds

		Number of combinations to find…	
N	**2^N**	**Globally Optimal**	**Best 2 Set Solution**
1	2	1	0
2	4	2	1
3	8	5	3
4	16	15	7
5	32	52	15
6	64	203	31
7	128	877	63
8	256	4,140	127
9	512	21,147	255
10	1,024	115,975	511
15	32,768	1.383 e+09	16,383

What if we *only* split into two sets?

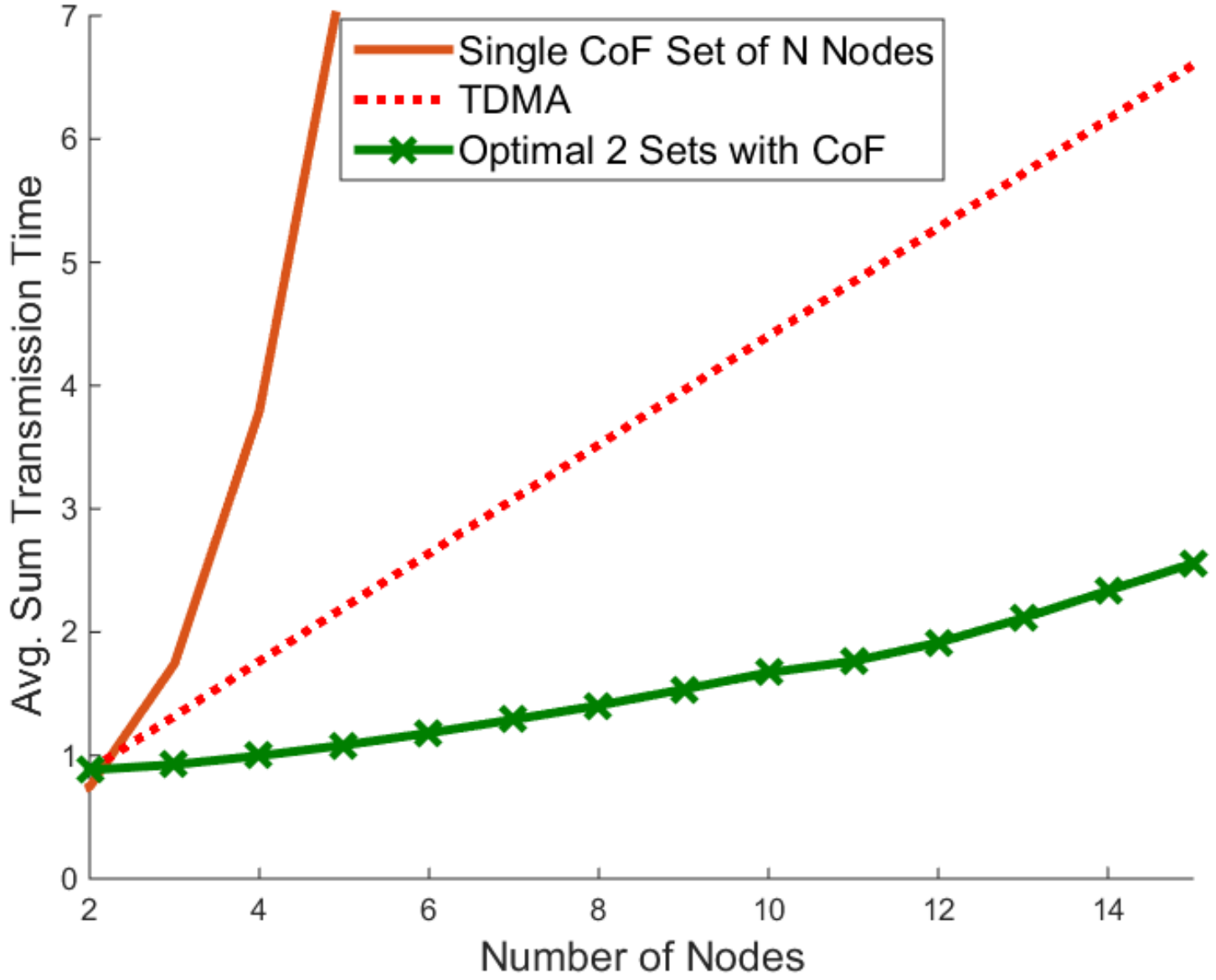
- $2^{N-1}-1$
- Brings us back to your typical combinatorial problem

Would this be better than a single set?

Limited Sets

- P=20dB
- h=0.3+C*N*(0,1)
- a_i=1

Sometimes more orthogonality is good



What if we *only* split into three sets?

		Number of combinations to find…		
N	**2^N**	**Globally Optimal**	**Best 2 Set Solution**	**Best 3 Set Solution**
1	2	1	0	0
2	4	2	1	0
3	8	5	3	1
4	16	15	7	6
5	32	52	15	25
6	64	203	31	90
7	128	877	63	301
8	256	4,140	127	966
9	512	21,147	255	3,025
10	1,024	115,975	511	9,330
15	32,768	1.383 e+09	16,383	2.375 e+06

- More combinations…

Do we want more orthogonal partitions?

***Yes** and more of it as N grows*

Simpler Solution for Fewer Computations

		Number of combinations to find…		
N	**2^N**	**Globally Optimal**	**Best 2 Set Solution**	**Sorted Solution**
1	2	1	0	0
2	4	2	1	2
3	8	5	3	5
4	16	15	7	8
5	32	52	15	12
6	64	203	31	16
7	128	877	63	20
8	256	4,140	127	24
9	512	21,147	255	29
10	1,024	115,975	511	34
15	32,768	1.383 e+09	16,383	59

- We know CoF behaves *nicely* if…
  - h similar to a
  - h_i similar to h_j

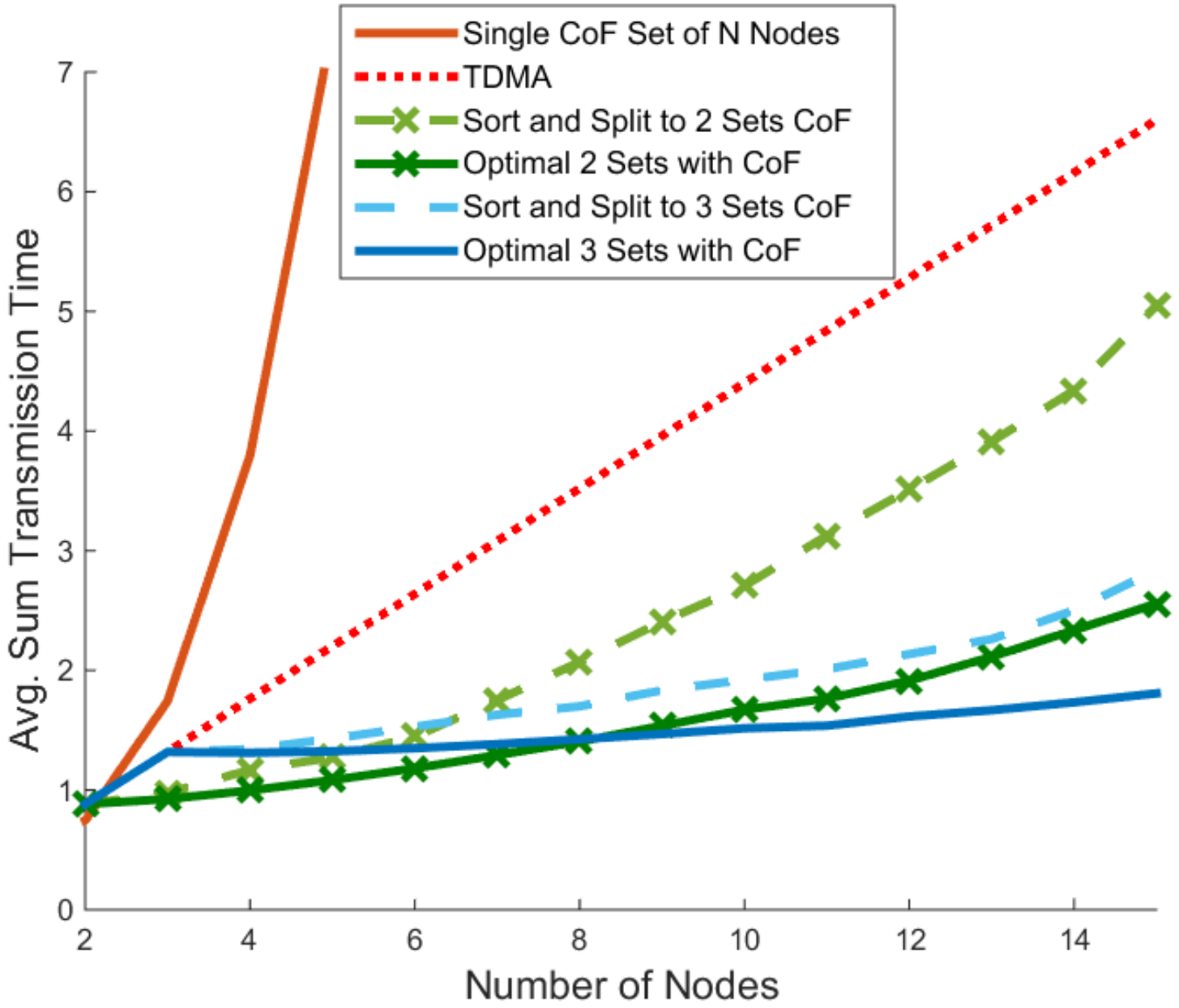
What if we simply sort then arbitrary split into groups?

$$h_i > h_j > h_k > h_l > h_m > h_N$$

Simplify

- P=20dB
- h=0.3+C$\mathcal{N}$(0,1)
- a_i=1

*Big drop in complexity, but **not** a big drop in performance*



Minimizing Time to Empty with CoF

Results

- Problem is hard in the NP sense
- At most N subsets in optimal solution
- Performance analysis of various strategies

Take home messages

- Simple heuristics can go a long way
- CoF can benefit from orthogonal partitions
- We haven't *yet* broken up with orthogonality