Repeated Games, Optimal Channel Capture, and Open Problems for Slotted Multiple Access

Michael J. Neely
(mikejneely@gmail.com)

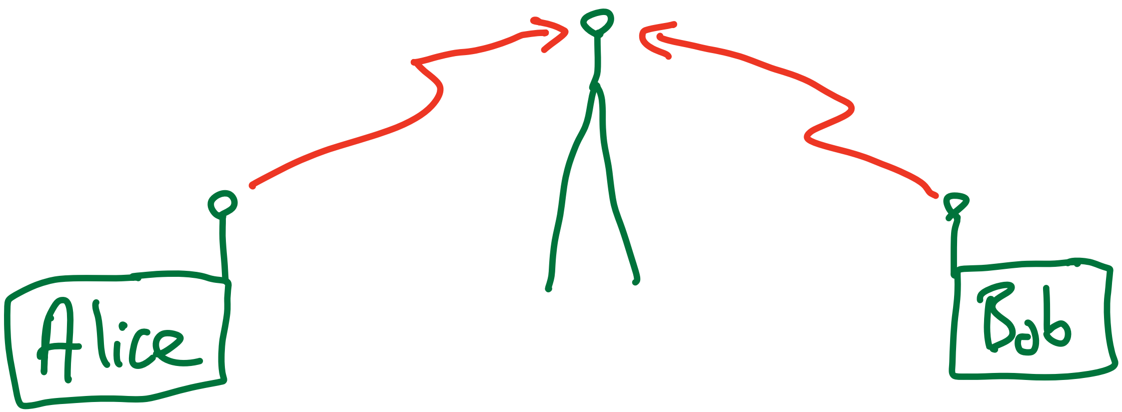
University of Southern California

Friday Nov. 5, 2021

Communication
Information
Learning
Quantum

Outline

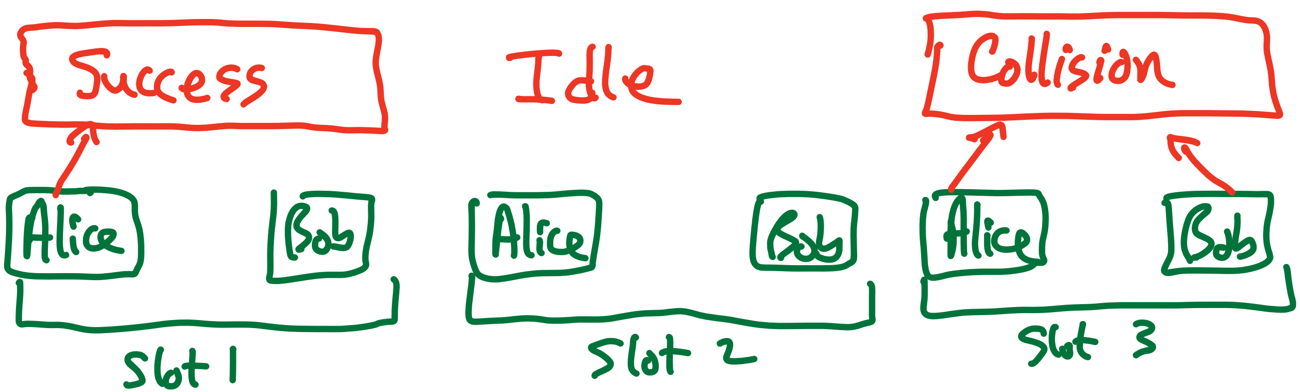
Multiple Access (MAC) refers to multiple network users that send data packets to a common access point (receiver).



1. EE 550 MAC Game Competition
2. Minimizing expected time to capture a channel

Part 1: EE 550 MAC Game Competition

- 2 users
- Each user has infinite number of fixed-length packets
- Time slots $t \in \{1, 2, 3, ..., T\}$ $(T = 100)$
- 1 packet transmission takes 1 time slot
- On each slot t: Do we transmit (1) or not (0)?
- Idle/Success/Collision



Game Structure for 1 slot

Player 1 \ Player 2	0	1
0	(0,0)	(0,1)
1	(1,0)	(0,0)

- Players 1 and 2
- Binary decisions: Transmit (1); Do not transmit (0)
- (Player *i* gets a point) iff (Player *i* transmits alone)

Binary decision similar to Prisoner Dilemma game (but different payoffs)

Volunteers to play a game?

Rules

1. Volunteers should turn on zoom video and put name in zoom id
2. 3 slots
3. Decentralized: Players cannot exchange secret chat messages.
4. Sequential decisions: On first slot the volunteers secretly write either 0 or 1 on paper. I then ask them to simultaneously reveal their decision in zoom video. Repeat procedure for slots 2 and 3.
5. Your grade is proportional to your score.

	Bhaskar	Ben	Outcome
→ 1	1	1	Collision
→ 2	1	1	Collision
3	1	1	Collision

No points scored

Student competition Fall 2021

1. 10 algorithms compete in 2-player 100-slot games:
   - 7 student algs, 1 instructor alg
   - AlwaysTransmit
   - NeverTransmit

     Important for alg to learn to use all resources when available
2. All algorithm pairs (i, j) compete $\forall i, j \in \{1, \ldots, 10\}$
   - Including (i, i), an independent version of same alg

     Important for alg to perform well against itself

3. Goal: Accumulate the most points over all ten 100-slot games that you play.

Matlab details

- Player subroutine: (use "persistent" local variables)

$$X = MyDecisionAlg(t, MyHistory[t], OpponentHistory[t]);$$

- Master Program:

for $t \in \{1, \ldots, 100\}$:

1. $X_1 = Player1DecisionAlg(t, Hist_1[t], Hist_2[t]);$
2. $X_2 = Player2DecisionAlg(t, Hist_2[t], Hist_1[t]);$
3. Tally scores;
4. Update history:

$$Hist_1[t] = [Hist_1[t]; X_1];$$

$$Hist_2[t] = [Hist_2[t]; X_2];$$

Some baseline algs

- AlwaysTransmit
- Tit-for-tat-1:
  1. Slot 1: $X[1] = 1$
  2. Slot $t \in \{2, \ldots, 100\}$: $X[t] = X_{opponent}[t-1]$
- Tit-for-tat-0:
  Same as Tit-for-tat-1 except $X[1] = 0$.
- 3-state
- 4-state
- 4-state with greedy ending

Figures of Merit

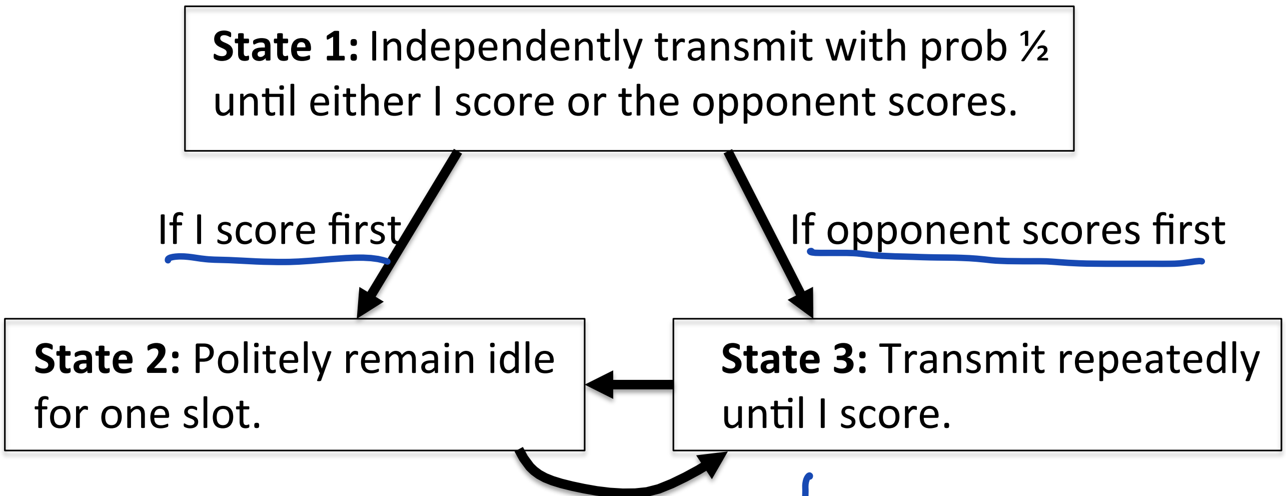
- SelfCompetition score α: What is your expected score when playing an independent version of yourself?
- NoCompetition score β: What is your expected score when playing NeverTransmit?
- HumanCompetition score γ: Simulated over 135 algs

Def: A *deterministic* algorithm uses no rand() calls.

Lemma: Every deterministic algorithm has $\alpha = 0$.

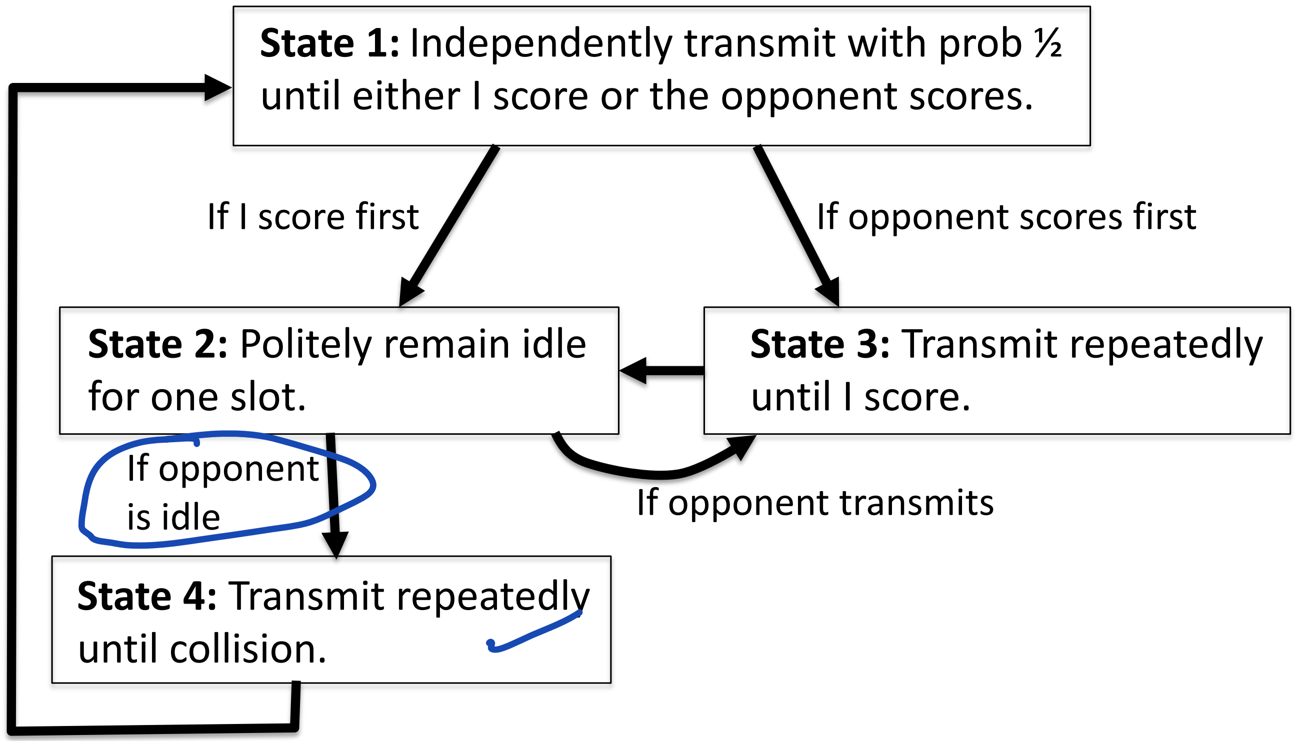
Tit-for-tat is deterministic.

3-state Alg



Punishing mech.

4-state Alg



State 1: Independently transmit with prob ½ until either I score or the opponent scores.
If I score first
If opponent scores first
State 2: Politely remain idle for one slot.
State 3: Transmit repeatedly until I score.
If opponent is idle
If opponent transmits
State 4: Transmit repeatedly until collision.

Fall 2021 Results

NT AT 4-state TFT1

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Totals	
A1	0	0	0	0	0	0	0	0	0	0	0	NT
A2	100	0	1	0	48.89	0	19.99	10.72	0	50.03	230.6444	AT
A3	98.03	0	49.49	0	49.41	49.33	21.61	24.9	0	33.27	326.0706	4-state
A4	39.94	0	1	0	20.46	0	20.06	10.69	0	36.27	128.4562	
A5	49.95	0	0.49	0	25.04	0	9.98	5.33	0	24.88	115.6895	
A6	1	0	49.67	0	0.5	0	19.05	31.93	0	25.26	127.4265	TFT1
A7	100	0	19.12	0.54	49.98	18.81	16.45	19.21	0.56	34.46	259.1658	
A8	50.52	0	23.85	0.02	25.34	31.57	18.65	24.74	0.19	27.37	202.2952	
A9	100	0	1	0	49.97	0	19.88	10.76	0	38.18	219.8058	
A10	50.01	0	16.68	13.75	25.15	24.76	15.61	22.73	11.81	24.98	205.5207	

Figure: Row i gives the points Alg A_i scored against each other alg. For each algorithm pair (i, j), the score is an average of 10^6 independent runs of separate 100-slot games.

- A1: NeverTransmit
- A2: AlwaysTransmit
- A3: 4-state
- A6: Tit-for-tat-1
- A7: Highest scoring student (Krishi)

Results by semester

	4-State	Second Place	AlwaysTransmit	AvgAlg
Fall 2021 (10 algs)	32.46	26.02	22.90	18.14
Fall 2020 (25 algs)	23.92	22.82	12.36	12.10
Fall 2019 (19 algs)	30.55	30.07	18.32	16.25
Spring 2018 (35 algs)	56.31	53.62	25.55	33.71
Fall 2018 (27 algs)	32.44	29.63	15.42	17.11
Spring 2017 (21 algs)	20.44	17.68	8.00	10.88
Fall 2016 (14 algs)	20.22	17.53	11.22	10.22

- Scores are presented as average score per (100-slot) game.
- 4-state came in 1st place every semester
- AvgAlg is the average score over all algs that semester.

Figures of Merit for main algorithms

	SelfComp α	NoComp β	Tournament γ
4-state	49.500	98.000	24.613
3-state	49.500	49.833	22.548
Tit-for-tat-0	0	0	20.410
Tit-for-tat-1	0	1	15.326
AlwaysTransmit	0	100	10.714

- 4-state was the top alg of 135 algs.
- The tournament score γ can be viewed as HumanCompetition score. It is the average score per game, based on simulation, when **competing against all 135 algorithms** designed over the 7 semesters.

Relate to Life Philosophy?

Competing mindsets on the ladder to success:

1. Rise up by pushing others down.
2. Help everyone around to rise up, including yourself.

The second mindset is consistent with particular results of this MAC game.

Math analysis for T-slot games

Theorem:

a) The SelfCompetition score for 4-state (and 3-state) is

$$\alpha = \frac{T-1}{2} + \left(\frac{1}{2}\right)^{T+1}$$

$T = 100 \implies \alpha \approx 49.500000000000000000000000000000394$

b) Converse: No algorithm that competes against an independent copy of itself can do better.

Quick proof of weaker converse

Weaker claim: Fix $T = 100$. Any algorithm that competes against an independent copy of itself has $\mathbb{E}[\mathit{SelfScore}] \leq 50$.

Proof: Let S_1 and S_2 be the scores of Players 1 and 2 at the end of 100 slots. Then...

$$S_1 + S_2 \leq 100$$

$$E[S_1] + E[S_2] \leq 100$$

$$E[S_1] = E[S_2]$$

$$\Longrightarrow \quad E[S_1] \leq 50.$$

Part 2: Expected time to capture channel

1. n users; slotted time
2. Everyone knows there are n
3. Users are indistinguishable (labels $\{1, 2, ..., n\}$ unknown)
4. Design an alg that is independently used by each user to minimize the expected time until the first success

Motivations for this problem

1. Max thruput.

   First user to succeed can capture channel indefinitely

2. Fair thruput.

   First user to succeed can capture channel for k more slots

3. Recursive construction of round robin schedule

4. Fundamental learning time needed to distinguish 1 user from n indistinguishable users

Related work

- Distributed control
  1. Witsenhausen 1973, 1987
     Proof for $n = 3$ agents; $n > 3$ open
  2. Nayyar and Teneketzis 2019
     Common Information

- Regret-based and online convex opt
  1. Bubeck and Budzinski 2020
  2. Bubeck, Li, Peres, Sellke 2020
  3. Kalathil, Nayyar, Jain 2014

- Distributed MAC, Poisson arrivals, Splitting and Tree Algs
  1. Bertsekas and Gallager 1992
  2. Mosely and Humblet 1985
  3. Tsybakov and Mikhailov 1978, 1980, 1981
  4. Hayes 1978
  5. Capetanakis 1979

Collision feedback $F[t]$

At end of each slot t, all users receive feedback:

$$\boxed{F[t] = \text{Number of users who transmitted}}$$

- $F[t] = 0$ (Idle)
- $F[t] = 1$ (Success and done)
- $F[t] = 2$ (Collision of 2 users)
- $F[t] = 3$ (Collision of 3 users)

  ...
- $F[t] = n$ (Collision of n users)

Detailed Collision

We can know $F[t]$ by, for example,

1. Measuring energy in collision
2. Using bit signature and counting spikes in matched filter [Gollakota and Katabi ZigZag 2008]

Proposed Alg for $n = 2$

Both users independently transmit with prob $1/2$ every slot until first success.

- $Z =$ random time to first success
- $\boxed{z_2 = \mathbb{E}[Z] = 2}$
- Converse: Cannot do better than 2

Proposed Alg for $n = 3$

Transmit with prob p and observe $F[t]$:

- $F[t] = 0$: Repeat
- $F[t] = 1$: Success. Done in 1 slot.
- $F[t] = 2$: $\cancel{\{1,2\}}$ $\{3\}$ Done in 2
- $F[t] = 3$: Repeat

Case $n = 3$ continued

$$\mathbb{E}[Z] = \overbrace{\mathbb{E}[Z|F[1]=0]}^{1+\mathbb{E}[Z]}(1-p)^3$$

$$+\overbrace{\mathbb{E}[Z|F[1]=1]}^{1}3p(1-p)^2$$

$$+\overbrace{\mathbb{E}[Z|F[1]=2]}^{2}3p^2(1-p)$$

$$+\overbrace{\mathbb{E}[Z|F[1]=3]}^{1+\mathbb{E}[Z]}p^3$$

Z = Random time to first success

Finish case $n = 3$

- Get:

$$\boxed{\mathbb{E}[Z] = \frac{1 + 3p^2(1-p)}{1 - p^3 - (1-p)^3}}$$

- Now optimize p:

$$z_3 = \inf_{p \in (0,1)} \left\{ \frac{1 + 3p^2(1-p)}{1 - p^3 - (1-p)^3} \right\}$$

$$\boxed{\begin{aligned} \implies p^* &= 0.411972 \\ z_3 &= 1.78795 \end{aligned}}$$

$z_3 < 2.$

Proposed Alg for general n

Transmit with prob p and observe $F[t]$:

- $F[t] = 0$: Repeat
- $F[t] = 1$: Done in 1
- $F[t] = k \in \{2, \ldots, n-2\}$:

  Choose better of groups: $\{1, \ldots, k\}, \{k+1, \ldots n\}$

- $F[t] = n-1$: Done in 2
- $F[t] = n$: Repeat

$$z_n = \inf_{p \in (0,1)} \left\{ \frac{1 + \sum_{i=2}^{n-1} \min\{z_i, z_{n-i}\} \binom{n}{i} p^i (1-p)^{n-i}}{1 - p^n - (1-p)^n} \right\}$$

Results

n	p_n	z_n	z_n^*
1	1	1	✓
2	0.5	2	✓
3	0.411972	1.78795	✓
4	0.302995	2.13454	✓
5	0.238640	2.15575	?
6	0.191461	2.26246	✓
7	0.166629	2.27543	?

- Conjecture: This algorithm is optimal for all $n \in \{1, 2, 3, \ldots\}$
- Have proof for special cases $n \in \{1, 2, 3, 4, 6\}$
  Other cases are open

Converse for case $n = 4$

- Imagine an optimal alg that achieves z_4^*.
- Let Z be random time to first success ($\mathbb{E}[Z] = z_4^*$)

Transmit on first slot with some probability p^* and observe $F[1]$:

$$E[Z \mid F[1]=0] \geq 1 + z^*$$

$$E[Z \mid F[1]=1] \geq 1$$

$$\boxed{E[Z \mid F[1]=2] \geq 2 \quad \leftarrow \text{Hard}}$$

$$E[Z \mid F[1]=3] \geq 2$$

$$E[Z \mid F[1]=4] \geq 1 + z^*$$

Why case $\{1,2\},\{3,4\}$ is hard

- Want to prove it is optimal to throw one group away
- Generally:
  1. Group $\{1,2\}$ can transmit next with prob q
  2. Group $\{3,4\}$ can transmit next with prob r
  3. Suppose we get feedback $F[2]=2$:

$$\{a,b\},\{c,d\}$$

- Exponentially growing (distributed) information state:
  1. User 1 history: $\{001101...\}$
  2. User 2 history: $\{110010...\}$
  3. User 3 history: $\{111001...\}$
  4. User 4 history: $\{111010...\}$

Proof idea

- Pesky case of $\{1, 2\}, \{3, 4\}$.
- Want to bound expected remaining time under any algorithm for this pesky case:

$$\mathbb{E}\left[R\right] \geq 2$$

- Consider new system with **2 virtual users with enhanced capabilities**.
- Show virtual system has $\mathbb{E}\left[R_{virtual}\right] \geq 2$
- Show virtual system can emulate the $\{1, 2\}, \{3, 4\}$ case.

2 virtual users with enhanced capabilities

> Every slot, each of the 2 indistinguishable virtual users can send any integer number of packets.

Proof of emulation of any algorithm on actual case $\{1,2\},\{3,4\}$

- ALG A: For actual group $\{1,2\}$
- ALG B: For actual group $\{3,4\}$

1. Over time, virtual user 1 independently simulates ALG A for user 1 and ALG B for user 3

   If both transmit then send 2 packets

   If only one transmits then send 1 packet

2. Over time, virtual user 2 independently simulates ALGA A for user 2 and ALG B for user 4

   If both transmit then send 2 packets

   If only one transmits then send 1 packet

Conclusions

1. MAC Game
   - Sharing is good. Greedy is bad.
   - Randomness is required
   - 4-state consistently wins competitions
     (and maximizes self-score α)

2. Time to first capture
   - Fundamental learning time needed to distinguish one user
   - Information state has complexity explosion
   - Optimality for $\boxed{n \in \{1, 2, 3, 4, 6\}}$
     (Novel method of virtual users with enhanced capabilities)

3. Open problems
   - $n = 5$ case; Cases $n \geq 7$
   - More limited forms of feedback is open
   - Multi-channel case
     (Arxiv paper explores case multi-channel 2-user and 3-user)

Related NSF grants

This talk

1. NSF SpecEES 1824418
   - M. J. Neely, "Repeated Games, Optimal Channel Capture, and Open Problems for Slotted Multiple Access," arXiv technical report, arXiv:2110.09638v1.
   - X. Zhou, I. Koprulu, A. Eryilmaz, M. J. Neely, "Low-Overhead Distributed MAC for Serving Dynamic Users over Multiple Channels," Proc. WiOpt 2021.
   - M. J. Neely, "Reversible Models for Wireless Multi-Channel Multiple Access," Proc. IEEE INFOCOM, 2021.
2. NSF CCF-1718477
   - M. J. Neely, "Fast Learning for Renewal Optimization in Online Task Scheduling," Journal of Machine Learning Research (JMLR) Sept. 2021.
   - M. J. Neely, "A Converse Result on Convergence Time for Opportunistic Wireless Scheduling," Proc. IEEE INFOCOM 2020.
   - K. Asgari and M. J. Neely, "Bregman-style Online Convex Optimization with Energy Harvesting Constraints," Proc. ACM Meas. Anal. Comput. Syst, Dec. 2020.