Using Proof-of-Presence to Coordinate

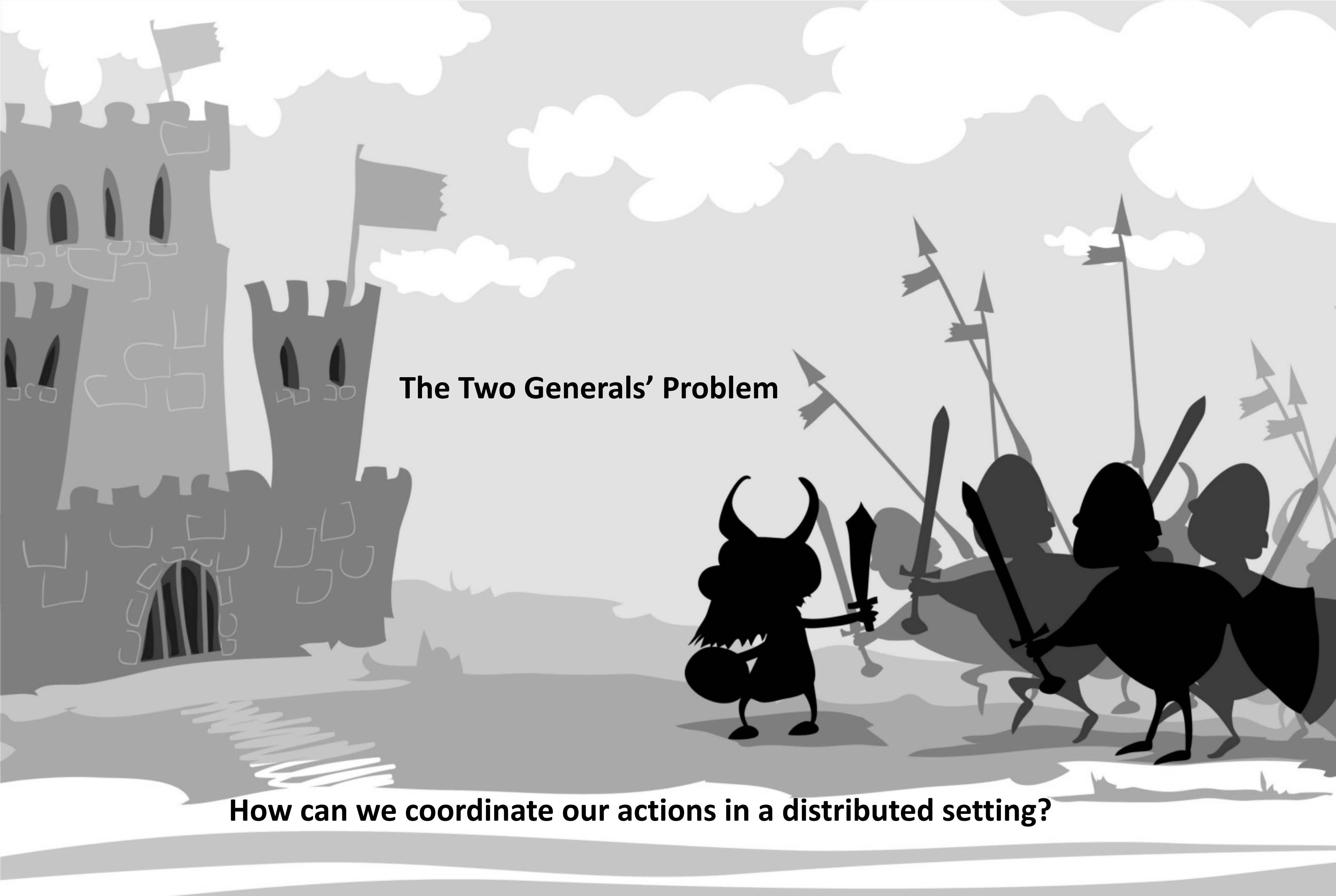
Adam Brandenburger* and Kai Steverson‡

* J.P. Valles Professor, NYU Stern School of Business
Distinguished Professor, NYU Tandon School of Engineering
Faculty Director, NYU Shanghai Program on Creativity + Innovation
Global Network Professor
New York University

‡ DCI Solutions

Version 06/14/21

The Two Generals’ Problem



How can we coordinate our actions in a distributed setting?

E. Akkoyunlu, K. Ekanadham, and R. Huber, “Some Constraints and Tradeoffs in the Design of Network Communications,” 1975; J. Gray, “Notes on Data Base Operating Systems,” 1977; image from https://medium.com/coinmonks/a-note-from-anthony-if-you-havent-already-please-read-the-article-gaining-clarity-on-key-787989107969

Game-Theory Perspective: A First Take

If messenger #1 arrives safely

then both generals know the plan is to attack at dawn

If messenger #2 arrives safely

then both generals know that both generals know the plan

If messenger #3 arrives safely

then both generals know that both generals know that both generals know the plan

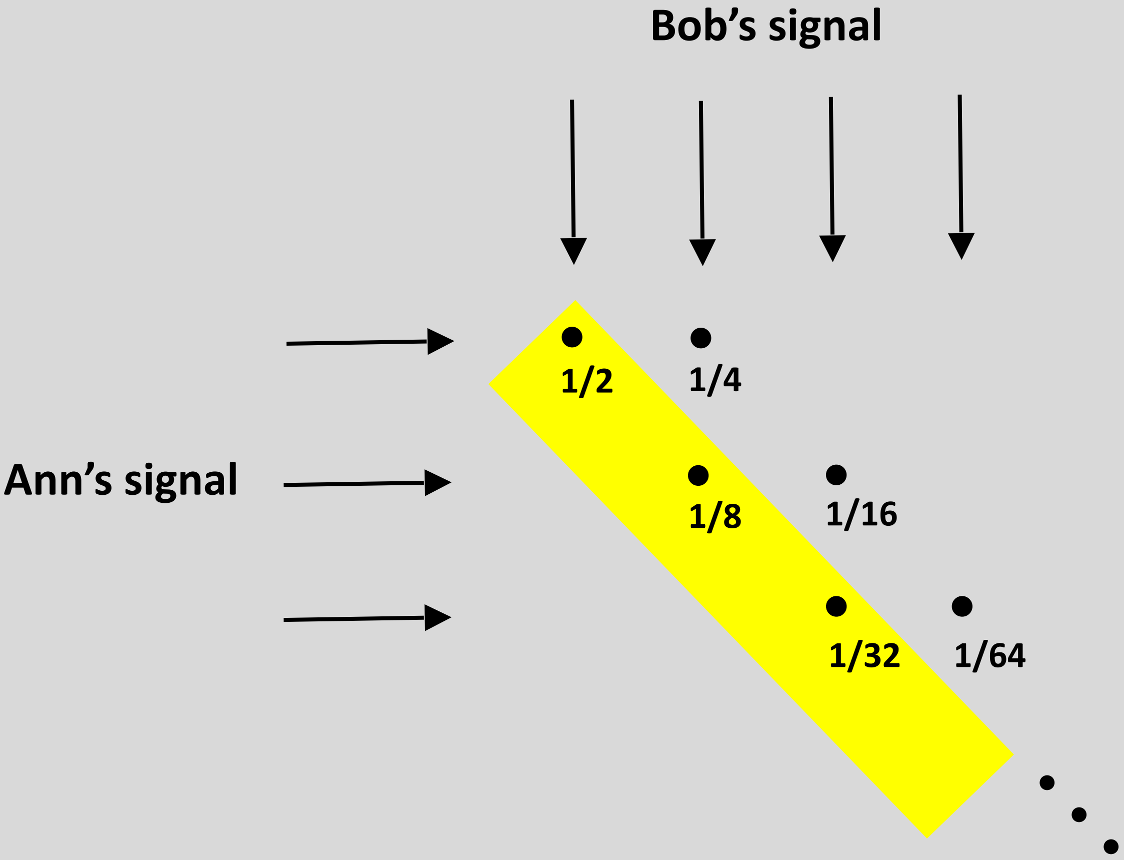
…

No finite sequence of messages will achieve common knowledge of the plan

Implicitly, the view is that if the generals could achieve common knowledge of the plan, then they would attack --- but that even high-order mutual knowledge of the plan does not suffice

Common Knowledge: A ‘Discontinuity at Infinity’

In game theory, the sensitivity of behavior to high-order mutual knowledge vs. common knowledge was first observed by Geanakoplos and Polemarchakis (1982) in the setting of the Agreement Theorem (Aumann, 1976)



J. Geanakoplos and H. Polemarchakis, “We Can’t Disagree Forever,” 1982; R. Aumann, “Agreeing to Disagree,” 1976; this variant is due to John Geanakoplos (private communication)

Additional Examples

Rubinstein (1989) formalized the inadequacy of high-order mutual knowledge in a version of the Two Generals Problem (with uncertainty over the payoff functions)

Aumann and Brandenburger (1995) showed that common knowledge of the players' conjectures in a game (in the presence of other assumptions) yields Nash equilibrium, but high-order mutual knowledge does not

A. Rubinstein, "The Electronic Mail Game: Strategic Behavior Under Almost Common Knowledge," 1989;
R. Aumann and A. Brandenburger, "Epistemic Conditions for Nash Equilibrium," 1995

Game-Theory Perspective: A Second Take

We next present another game-theory formulation of the Two Generals Problem

Now there is uncertainty over the number of generals who are present

Each general might be confident about the attack time but less confident about how many other generals will be present to attack

Is there a way to achieve a reliable (albeit probabilistic) headcount?

We provide a positive answer (inspired by the idea of proof-of-work*) via a mechanism we call "proof of presence"

* [I]n a POW, a prover demonstrates to a verifier that she has performed a certain amount of computational work in a specified interval of time

* M. Jakobsson and A. Juels, "Proofs of Work and Bread Pudding Protocols," in B. Preneel (ed.), *Secure Information Network*s, Kluwer Academic Publishers, 1999, pp. 258–272

A Coordination Game with an Uncertain Number of Active Players

Players may be inactive (effectively choosing ∅) or active (choosing *c* or ∅)

Coordination is positive only if there is a sufficient (expected) number of active players

	c	∅
c	3α-1 3α-1 3α-1	2α-1 0 2α-1
∅	0 2α-1 2α-1	0 0 α-1

c

	c	∅
c	2α-1 2α-1 0	α-1 0 0
∅	0 α-1 0	0 0 0

∅

The idea is that action *c* will be chosen if and only if

α × expected number of active players ≥ 1

Adding a Computational Puzzle to the Game

A computational puzzle is posted to a message board at time 0

Each active player has a machine that works on the puzzle and finds the solution with Poisson arrival rate λ (independent across machines)

If a machine solves the puzzle, there is a delay until time *T*, when the solution is posted to the board (otherwise a null message is posted)

The puzzle can be solved only by guesswork but the solution can be immediately verified

This procedure (“proof of presence”) is inspired by:

> “Every general, just by verifying the difficulty of the proof-of-work chain, can estimate how much parallel CPU power per hour was expended on it and see that it must have required the majority of the computers to produce that much proof-of-work in the allotted time” *

* http://satoshi.nakamotoinstitute.org/emails/cryptography/11/; see also
S. DeDeo, “The Bitcoin Paradox: Why Cryptocurrency Will Always Be Political,” 2017, at
http://nautil.us/issue/55/trust/the-bitcoin-paradox

Probability Calculations

The probability that *k* players are active, conditional on a solution by time *T*, is given by

$$\phi(k;T) = \frac{p_k\left[1 - \exp\left(-\lambda kT\right)\right]}{\sum_{i=1}^{n} p_i\left[1 - \exp\left(-\lambda iT\right)\right]}$$

We are interested in cases where

$$\alpha \sum_{k=1}^{n} k \cdot p_k < 1$$

but there is a (finite) *T* such that

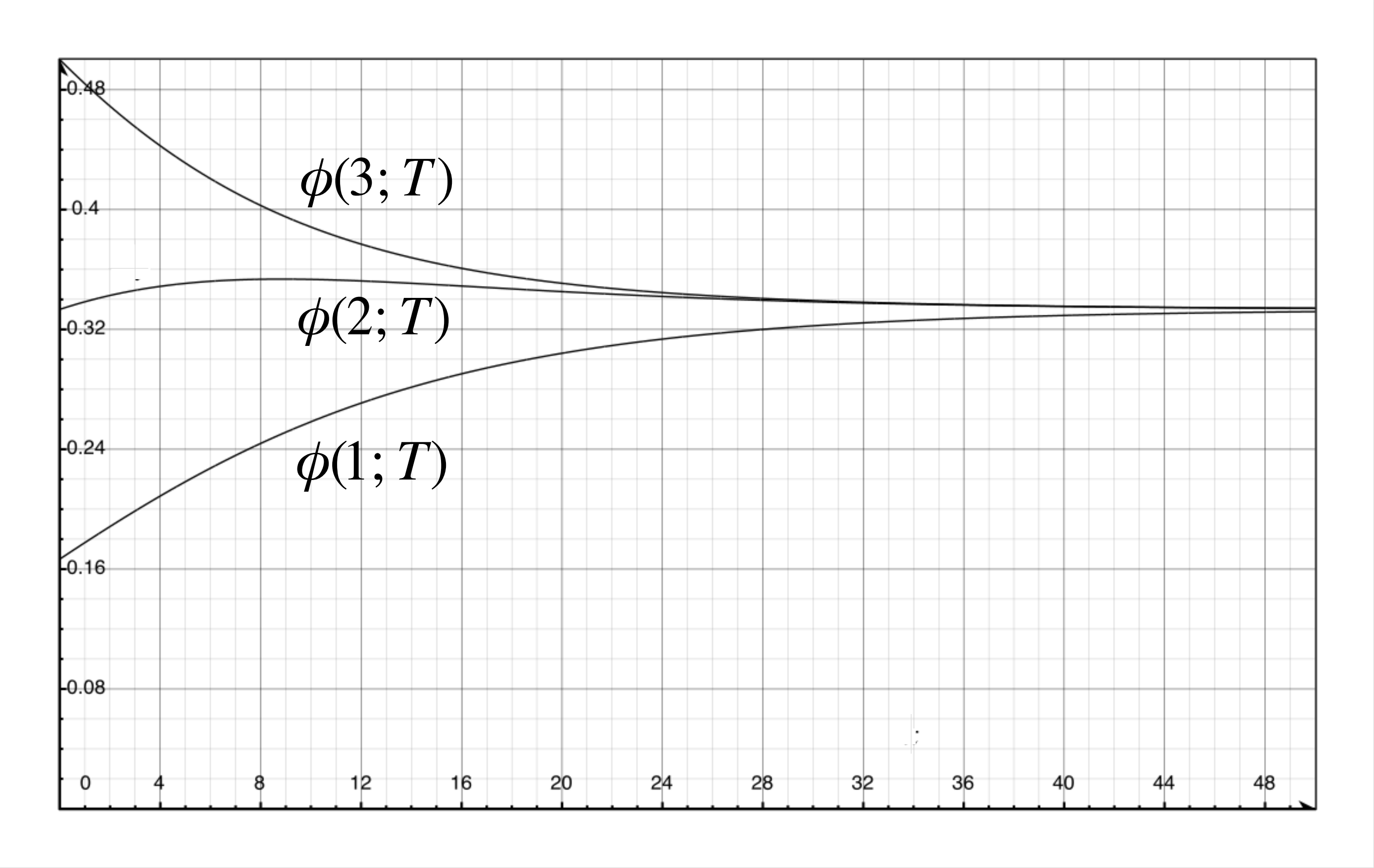
$$\alpha \sum_{k=1}^{n} k \cdot \phi(k;T) \geq 1$$

The idea is that we can choose a time *T* so that, if a solution is found by *T*, then there is a good chance that a good number of players are active

Calculations with Three Players

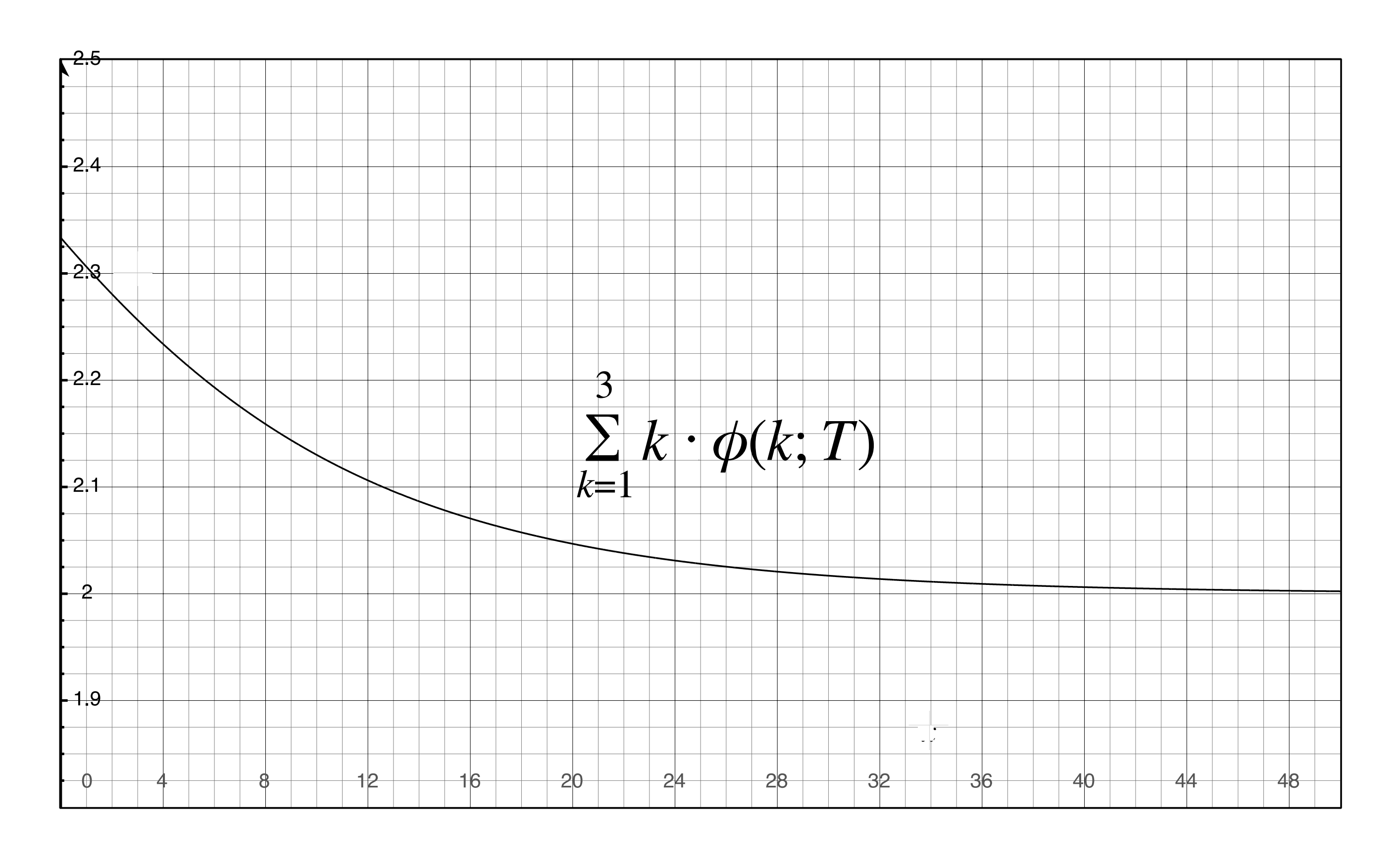
# of players *n* = 3; arrival rate λ = 0.1; uniform prior on # of active players

Probability that *k* players are active, conditional on a solution by *T*:



Calculations with Three Players contd.

Expected number of active players, conditional on a solution by *T*:



Proposition

The graph validates our earlier intuition that a rapid solution to the computational puzzle should make choosing the coordination action more attractive to players

If

$$\frac{\sum_{k=1}^{n} k\, p_k}{\sum_{k=1}^{n} k^2\, p_k} < \alpha < \frac{1}{\sum_{k=1}^{n} k\, p_k}$$

then coordination does not happen without the computational puzzle but can happen, for sufficiently small T, with the computational puzzle

Discussion

We can allow for malicious agents who participate in solving the puzzle but not in the game (treated in the full paper)

We conjecture we could vary the arrival rates (the λ's) by player to model different levels of computational power, without qualitatively changing the results

We might want to include rewards for solving the puzzle, to avoid free-riding where players do not work on the puzzle but join the coordination game

We have not looked at the question of the optimal threshold *T*, which balances various factors