SHORT-PAPER

Brief Announcement: Rise and Shine Efficiently! The Complexity of Adversarial Wake-up

PETER ROBINSON, Augusta University, Augusta, GA, United States

MING MING TAN, Augusta University, Augusta, GA, United States

Published: 13 June 2025

PODC '25: ACM Symposium on Principles of Distributed Computing
June 16 - 20, 2025
Huatulco, Mexico

Conference Sponsors:
SIGOPS
SIGACT

PODC '25: Proceedings of the ACM Symposium on Principles of Distributed Computing (June 2025)
https://doi.org/10.1145/3732772.3733504
ISBN: 9798400718854

Brief Announcement: Rise and Shine Efficiently! The Complexity of Adversarial Wake-up

Peter Robinson
Augusta University
Augusta, Georgia, USA
perobinson@augusta.edu

Ming Ming Tan
Augusta University
Augusta, Georgia, USA
mtan@augusta.edu

ABSTRACT

We study the wake-up problem in distributed networks, where an adversary awakens a subset of nodes at arbitrary times, and the goal is to wake up all other nodes as quickly as possible by sending only few messages. We prove the following lower bounds:

- We first consider the setting where each node receives advice from an oracle who can observe the entire network, but does not know which nodes are awake initially. More specifically, we consider the KT_0 LOCAL model with advice, where the nodes have no prior knowledge of their neighbors. We prove that any randomized algorithm must send $\Omega\left(\frac{n^2}{2^\beta \log n}\right)$ messages if nodes receive only $O(\beta)$ bits of advice on average.
- For the KT_1 assumption, where each node knows its neighbors' IDs from the start, we show that any $(k+1)$-time algorithm requires $\Omega\left(n^{1+1/k}\right)$ messages. Our result is the first super-linear (in n) lower bound, for a problem that does not require individual nodes to learn a large amount of information about the network topology, which may be of independent interest. This applies even to the synchronous KT_1 LOCAL model, where the computation is structured into rounds and messages can be of unbounded length.

To complement our lower bound results, we present several new algorithms:

- We give an asynchronous KT_1 LOCAL algorithm that solves the wake-up problem with a time and message complexity of $O(n \log n)$ with high probability.
- We introduce the notion of *awake distance* ρ_{awk}, which is upper-bounded by the network diameter, and present a synchronous KT_1 LOCAL algorithm that takes $O(\rho_{awk})$ rounds and sends $O\left(n^{3/2}\sqrt{\log n}\right)$ messages with high probability.
- We give deterministic advising schemes in the asynchronous KT_0 $\mathsf{CONGEST}$ model (with advice). In particular, we obtain an $O\left(\rho_{awk} \log^2 n\right)$-time advising scheme that sends $O\left(n \log^2 n\right)$ messages, while requiring $O\left(\log^2 n\right)$ bits of advice per node.

PODC '25, June 16–20, 2025, Huatulco, Mexico

ACM ISBN 979-8-4007-1885-4/25/06.
https://doi.org/10.1145/3732772.3733504

CCS CONCEPTS

• **Theory of computation → Distributed algorithms**.

KEYWORDS

distributed wake-up, message complexity lower bound, computing with advice, randomization

ACM Reference Format:
Peter Robinson and Ming Ming Tan. 2025. Brief Announcement: Rise and Shine Efficiently! The Complexity of Adversarial Wake-up. In *ACM Symposium on Principles of Distributed Computing (PODC '25), June 16–20, 2025, Huatulco, Mexico.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3732772.3733504

1 INTRODUCTION

In this paper, we study distributed algorithms in networks where every node is either awake or asleep, and our goal is to understand whether it is possible to wake up all sleeping nodes efficiently. Our investigation is motivated by well-established networking standards such as Wake-on-LAN and Wake-on-Wireless-LAN [1, 2], where a sleeping node only listens to wake-up messages (called "magic packets") on its network adapter and does not perform any other computation. Allowing unused nodes to be in a sleep state can potentially lead to significantly reduced energy consumption in large networks and may increase the performance-per-watt ratio of data centers; e.g., see [9].

To formalize the setting, we consider an asynchronous communication network represented by a graph of n nodes and m edges. Each node is running an instance of a distributed algorithm by exchanging messages over its incident communication links (i.e., edges) of the network. Initially, an arbitrary non-empty subset of the nodes are awake, whereas all other nodes continue to sleep until they are woken up, e.g., by receiving a message from some already awake node. Two key metrics of a distributed algorithm are its time complexity and its message complexity. The former quantifies the worst case number of time units until the algorithm has reached its goal, whereas the latter counts the total number of messages sent throughout the execution. In this work, we investigate the time and message complexity of the *wake-up problem*, which was formally defined in [8]. That is, an adversary chooses a network topology and decides which nodes it wakes up and at what time, while the algorithm's goal is to wake up all other nodes as quickly and by sending as few messages as possible.

While it is clear that, without further assumptions, any algorithm must take time steps that is at least as large as the network diameter D for solving the wake-up problem, determining the achievable message complexity is less obvious, as prior work in this area suggests that the initial knowledge of the nodes can affect the achievable bounds significantly. Two well-studied assumptions are known as

KT_1 [4] and KT_0 [16]: The KT_1 assumption stipulates that each node starts out knowing who else it is connected to, which is an adequate abstraction for modern-day IP networks. On the other hand, KT_0 (also known as the *port numbering model*) requires that each node sends messages to its neighbors by using integer port numbers that are unrelated to the neighbors' IDs. Given that nodes have significantly less knowledge under the KT_0 assumption, it is not too surprising that sending a number of messages that is proportional to the number of edges in the network is a requirement for solving virtually any problems of interest (see [13]), and this easily extends to the wake-up problem. One possible way to circumvent this lower bound is to consider the KT_1 assumption, which allows breaking the KT_0 message complexity barrier for problems such as constructing a minimum spanning tree (MST). However, understanding the time and message complexity of the fundamental wake-up problem is still unresolved in networks under the KT_1 assumption. Our work takes a first step towards closing these gaps in literature.

We emphasize that for KT_0, v has no prior knowledge of the concrete mapping and that we assume that the other endpoint of an edge also learns the port connection, if one of its two vertices sends a message over its respective port.

The KT_1 *assumption* provides more knowledge to the algorithm than KT_0, since each node starts out knowing the IDs of all its neighbors, which it can use when sending messages. It is well known, see e.g., [10, 12] that KT_1 is powerful enough for implementing graph sketching techniques [3].

Adversary. An adversary determines the network topology, the node IDs, and the set of initially awake nodes. In the case of KT_0, the adversary also determines each individual node's port mapping. Apart from controlling the message delays, it may also decide to wake up a currently-sleeping node at any point in the execution.[1] When considering randomized algorithms, we assume that the adversary is *oblivious*, in the sense that it must decide the delay of in-transit messages and which nodes to wake up (and at what time) without knowing the state of the nodes, which includes their private random bits.

Awake Distance. In many real-world applications, it may be desirable to ensure that nodes wake up sooner than time proportional to the network diameter, if there are awake nodes located closer to them. This motivates introducing a more fine-grained way of quantifying the performance of an algorithm with respect to time. Given a graph G and a set of initially-awake nodes A_0 that are being awoken by the adversary, we define the *awake distance* $\rho_{awk} = \rho_{awk}(G, A_0) = \max_{u \in G} \text{dist}_G(A_0, u)$, where $\text{dist}_G(A_0, u)$ is the shortest hop distance of u to some node in A_0. Note that ρ_{awk} is equivalent to the time complexity of the (message-inefficient) standard flooding algorithm. As elaborated in more detail in Section 1.1 and summarized in Table 1, we have designed wake-up algorithms that send few messages and achieve a time complexity proportional to ρ_{awk}.

[1]This is a crucial difference to related work in the context of energy and awake complexity (e.g., see [5, 6]), where the assumption is that the algorithm (and not the adversary) controls the wake up schedule.

1.1 Contributions and Technical Challenges

We present several novel algorithms and lower bounds that demonstrate the interplay between time, messages, and length of advice. Table 1 summarizes our results.

1.1.1 *A Lower Bound on the Advice for Randomized Algorithms in* KT_0.

The work of [8] gives an elegant combinatorial argument that the total advice assigned to the nodes must be $\Omega(n \log n)$ bits, for obtaining a message complexity of $O(n)$ when assuming KT_0. However, their argument only holds for deterministic algorithms and does not reveal the actual message complexity required, e.g., when restricting the advice to $o(\log n)$ bits per node, i.e., $o(n \log n)$ bits in total.

We present a new result under the KT_0 assumption that not only works for randomized algorithms with advice, but also results in a polynomial improvement on the lower bound on the message complexity, when the advice provided to each node is small.

Theorem 1. *Let $\mathcal{A}$ be a randomized advising scheme that solves the wake-up problem in the (synchronous or asynchronous)* KT_0 LOCAL *model and errs with probability $\epsilon < \frac{1}{2\log_2 n}$. For any positive $\beta \leq \log_2 n$, if the expected message complexity of $\mathcal{A}$ is at most $\frac{n^2}{2^{\beta+4}\log_2 n}$, then the average length of advice per node is at least $\frac{1}{6} \cdot (\beta - 2 - o(1)) = \Omega(\beta)$ bits. In particular, an advice length of $o(\log n)$ bits per node requires an expected message complexity of $\Omega(n^{2-\alpha})$, for any constant $\alpha > 0$. This holds even if the oracle knows the set of awake nodes and even if we assume shared randomness.*

For proving Theorem 1, we define a lower bound graph $\mathcal{G}$, where every node v_i in a large set of the nodes (called center nodes) has exactly one edge to a sleeping neighbor w_i, who cannot be woken up by anyone else. Considering that nodes do not know their port mappings, it is not too difficult to show that the center nodes would essentially need to send messages across most of their ports, if we assumed the standard KT_0 LOCAL model without advice. The main technical challenge in proving Theorem 1 emanates from the fact that the oracle gets to see all port mappings when computing the advice. For instance, consider the special case where the message complexity is $O(n^{2-\alpha})$, for some small constant $\alpha > 0$. We need to take into account the possibility that the oracle encodes the $O(\log n)$ bits required for representing the port number leading to v_i's sleeping neighbor w_i. For instance, the oracle could partition the port number for w_i into $\omega(1)$ pieces and store each piece among a subset of the neighbors of v_i. This would, in fact, suffice for v_i identifying (and waking up) w_i, as it can receive messages from $\omega(\log n)$ of its neighbors, each of arbitrary size, without violating the message complexity bound. To avoid this pitfall, our proof leverages that there are many such nodes that each need to find their sleeping neighbor and, consequently, the oracle cannot successfully encode *all* of these ports among the nodes in the network without using at least $\Omega(n \log n)$ bits in total.

1.1.2 *A Lower Bound on the Message Complexity in* KT_1.

We also consider the challenging KT_1 setting (without advice), where nodes start out knowing their neighbors' IDs. In other words, the knowledge of the IDs associated with an edge is shared between its two endpoints, and this enables the use of powerful graph sketching techniques [3, 11, 12] that allow discovering "outgoing" edges (such as the edge $\{v_i, w_i\}$) with a polylogarithmic overhead of messages, as elaborated in Section 1.1. Furthermore, as shown in [17], graph

Time	Messages	Advice	Model	Random
Algorithms				
$O(n \log n)^*$	$O(n \log n)^*$	-	async. KT$_1$ LOCAL	yes
$O(\rho_{awk})$	$O(n^{3/2}\sqrt{\log n})^*$	-	*sync.* KT$_1$ LOCAL	yes
$O(D)$	$O\left(n^{3/2}\right)$	$O(\sqrt{n} \log n)$	async. KT$_0$ CONGEST	no
$O(D \log n)$	$O(n)$	$O(\log n)$	async. KT$_0$ CONGEST	no
$O(k\, \rho_{awk} \log n)$	$O\left(k\, n^{1+1/k}\right)$	$O\left(n^{1/k} \log^2 n\right)$	async. KT$_0$ CONGEST	no
$O\left(\rho_{awk} \log^2 n\right)$	$O\left(n \log^2 n\right)$	$O\left(\log^2 n\right)$	async. KT$_0$ CONGEST	no
Lower Bounds				
-	$\leq \frac{n^2}{2^{\beta+4} \log_2 n}$	$\Omega(\beta)$	*sync.* KT$_0$ LOCAL	yes
$\leq k+1$	$\Omega\left(n^{1+1/k}\right)$	-	*sync.* KT$_1$ LOCAL	yes

* With high probability.

Table 1: Algorithms and Lower Bounds for the Wake-up Problem. Column "Advice" refers to the maximum length of advice per node, unless stated otherwise. Column "Random" indicates whether nodes have access to random bits. Parameter ρ_{awk} refers to the awake distance.

sketching allows solving *any* graph problem with just $\tilde{O}(n)$ message complexity in the synchronous KT$_1$ CONGEST model, albeit at the expense of a prohibitively large time complexity, which stands in stark contrast to the unconditional lower bound of $\Omega(m)$ known to hold for the KT$_0$ assumption (without advice). Thus, we focus on *time-restricted* algorithms and present the first trade-off on the time and achievable message complexity for the wake-up problem under the KT$_1$ assumption.

Theorem 2. *Consider any integer $k \in [3, o(\log n)]$, and let $\mathcal{A}$ be a randomized Las Vegas algorithm that solves the wake-up problem in the (synchronous or asynchronous)* KT$_1$ LOCAL *model. If $\mathcal{A}$ takes at most $k+1$ units of time in every execution with awake distance $\rho_{awk} = 1$, then the expected message complexity of $\mathcal{A}$ is at least $\Omega\left(n^{1+1/k}\right)$.*

When disregarding the message complexity, a straightforward flooding algorithm solves the awake problem in optimal ρ_{awk} time. Thus, an immediate corollary of Theorem 2 is that optimality cannot be achieved in both, time and messages, under the KT$_1$ assumption, even if we allow unbounded messages and assume synchronous rounds.

We point out that, prior to this work, almost no lower bounds on the message complexity were known for problems in the general KT$_1$ setting. With the exception of [17], all previous lower bounds (e.g., [4, 7, 14, 15]) use edge-crossing arguments that only hold for comparison-based algorithms, which restrict nodes to behave the same when observing "order-equivalent" IDs among their neighbors. While the lower bound for graph spanners in [17] does hold for general algorithms under the KT$_1$ assumption, it exploits the fact that many nodes needs to identify a large set of incident edges to correctly compute a spanner, and it is unclear how to extend this approach to problems where the output size at individual nodes is negligible, as is the case for the wake-up problem. To the best of our knowledge, Theorem 2 is the first super-linear lower bound on the message complexity that holds for general KT$_1$ algorithms for a problem that does not require most nodes to learn a large amount of information about the graph topology. Moreover, our lower bound holds even in the LOCAL model, whereas all of the above-mentioned works use a bottleneck argument that crucially exploits the CONGEST model.

For proving Theorem 2, we modify the lower bound construction $\mathcal{G}$ used for KT$_0$ to define a new class of graphs $\mathcal{G}_k$ with large girth, while retaining the property that every node v_i in a certain set has exactly one *crucial neighbor* w_i among its $\Theta(n^{1/k})$ neighbors, who cannot be woken up by any other node. Our overall goal is to argue that identifying this one sleeping neighbor is hard, similarly as for KT$_0$. However, there are several technical challenges that we need to overcome:

- From the perspective of a node $v_i \in V$, it takes only $\sigma = O\left(\frac{\log n}{k}\right)$ bits of information for determining which of its $\Theta(n^{1/k})$ incident edges leads to its crucial neighbor w_i. Thus, at first glance, it might seem that the graph sketching techniques introduced by Ahn, Guha, and McGregor [3], which is known to yield $O(n \operatorname{poly} \log n)$ message complexity for spanning tree construction (see [12]), could be leveraged to wake-up every node with only a polylogarithmic message overhead.
- Note that we want to show that the result holds even in the KT$_1$ LOCAL model, where a node $v_i \in V$ may receive a polynomial number of bits during the execution. In particular, this means we cannot utilize a "bottleneck argument" based on small cuts to argue that v_i does not learn enough information about w_i quickly enough, as is commonly done in the CONGEST model (see, e.g., [19]).

1.1.3 Upper Bounds. We defer the discussion of our algorithms to the full paper [18] .

ACKNOWLEDGMENTS

Peter Robinson was supported in part by the National Science Foundation (NSF) grant CCF-2402836. Ming Ming Tan was supported in part by the National Science Foundation (NSF) grant CCF-2348346.

REFERENCES

[1] 2024. Wake-on-LAN — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Wake-on-LAN https://en.wikipedia.org/wiki/Wake-on-LAN. Accessed: 2024-10-11.

[2] Advanced Micro Devices, Inc. (AMD). 1995. *Magic Packet Technology*. Technical Report. https://www.amd.com/content/dam/amd/en/documents/archived-tech-docs/white-papers/20213.pdf https://www.amd.com/content/dam/amd/en/documents/archived-tech-docs/white-papers/20213.pdf. Accessed: 2024-10-11.

[3] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. 2012. Analyzing graph structure via linear measurements. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, 459–467.

[4] Baruch Awerbuch, Oded Goldreich, David Peleg, and Ronen Vainish. 1990. A Trade-Off between Information and Communication in Broadcast Protocols. *J. ACM* 37, 2 (1990), 238–256. https://doi.org/10.1145/77600.77618

[5] Yi-Jun Chang, Varsha Dani, Thomas P. Hayes, Qizheng He, Wenzheng Li, and Seth Pettie. 2018. The Energy Complexity of Broadcast. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, PODC 2018, Egham, United Kingdom, July 23-27, 2018*, Calvin Newport and Idit Keidar (Eds.). ACM, 95–104. https://doi.org/10.1145/3212734.3212774

[6] Soumyottam Chatterjee, Robert Gmyr, and Gopal Pandurangan. 2020. Sleeping is Efficient: MIS in $O(1)$-rounds Node-averaged Awake Complexity. In *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, Yuval Emek and Christian Cachin (Eds.). ACM, 99–108. https://doi.org/10.1145/3382734.3405718

[7] Fabien Dufoulon, Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, and Peter Robinson. 2024. The Message Complexity of Distributed Graph Optimization. In *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA (LIPIcs, Vol. 287)*, Venkatesan Guruswami (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 41:1–41:26. https://doi.org/10.4230/LIPICS.ITCS.2024.41

[8] Pierre Fraigniaud, David Ilcinkas, and Andrzej Pelc. 2006. Oracle size: a new measure of difficulty for communication tasks. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*. 179–187.

[9] Anshul Gandhi, Mor Harchol-Balter, and Michael A Kozuch. 2012. Are sleep states effective in data centers?. In *2012 international green computing conference (IGCC)*. IEEE, 1–10.

[10] James W Hegeman, Gopal Pandurangan, Sriram V Pemmaraju, Vivek B Sardeshmukh, and Michele Scquizzato. 2015. Toward optimal bounds in the congested clique: Graph connectivity and MST. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*. 91–100.

[11] Bruce M Kapron, Valerie King, and Ben Mountjoy. 2013. Dynamic graph connectivity in polylogarithmic worst case time. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 1131–1142.

[12] Valerie King, Shay Kutten, and Mikkel Thorup. 2015. Construction and Impromptu Repair of an MST in a Distributed Network with o(m) Communication. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, Chryssis Georgiou and Paul G. Spirakis (Eds.). ACM, 71–80. https://doi.org/10.1145/2767386.2767405

[13] Shay Kutten, Gopal Pandurangan, David Peleg, Peter Robinson, and Amitabh Trehan. 2015. On the Complexity of Universal Leader Election. *J. ACM* 62, 1 (2015), 7:1–7:27. https://doi.org/10.1145/2699440

[14] Shay Kutten, Peter Robinson, and Ming Ming Tan. 2024. Tight Bounds on the Message Complexity of Distributed Tree Verification. *arXiv preprint arXiv:2401.11991* (2024).

[15] Shreyas Pai, Gopal Pandurangan, Sriram V Pemmaraju, and Peter Robinson. 2021. Can We Break Symmetry with o (m) Communication?. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*. 247–257.

[16] David Peleg. 2000. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, Philadelphia.

[17] Peter Robinson. 2021. Being Fast Means Being Chatty: The Local Information Cost of Graph Spanners. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, Dániel Marx (Ed.). SIAM, 2105–2120. https://doi.org/10.1137/1.9781611976465.126

[18] Peter Robinson and Ming Ming Tan. 2024. Rise and Shine Efficiently! The Complexity of Adversarial Wake-up in Asynchronous Networks. *arXiv preprint arXiv:2410.09980* (2024).

[19] Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. 2012. Distributed Verification and Hardness of Distributed Approximation. *SIAM J. Comput.* 41, 5 (2012), 1235–1265. https://doi.org/10.1137/11085178X