On the Complexity of Fundamental Problems in Dynamic Ad-hoc Networks

SEBASTIAN ABSHOFF

University of Paderborn

Abstract: This thesis studies the complexity of fundamental problems in dynamic, i.e., time-variant, ad-hoc networks. Based on the model by Kuhn et al. (Symposium on Theory of Computing 2010), the network is controlled by an adaptive adversary that tries to prevent the efficient execution of algorithms and only guarantees connectivity in each round. In this thesis, three main aspects are considered, which can be found in three different parts of the thesis. In the first part, the adversary is restricted geometrically and an information dissemination problem is analyzed. The second part focusses on the counting problem (How many nodes are there in the network?) and establishes a relation to information dissemination problems. Finally, the third part studies the continuous, i.e., the repeated, computation of aggregation functions (e.g., the maximum of all inputs given to all nodes) in more stable variants of dynamic networks.

Zusammenfassung: Diese Arbeit beschäftigt sich mit Fragestellungen zur Komplexität grundlegender Probleme in dynamischen, d. h. zeitlich veränderlichen, Ad-hoc-Netzen. Basierend auf dem Model von Kuhn et al. (Symposium on Theory of Computing 2010) wird das Netz unter die Kontrolle eines adaptiven Gegenspielers gestellt, der versucht, die effiziente Ausführung von verteilten Algorithmen zu verhindern, und lediglich Zusammenhang in jeder Runde gewährleistet. In dieser Arbeit werden drei wesentliche Aspekte betrachtet, die sich in drei Teilen der Arbeit wiederfinden: Im ersten Teil wird der Gegenspieler zusätzlich geometrisch eingeschränkt und das Verbreiten von Informationen als grundlegendes Problem untersucht. Im zweiten Teil wird die Frage nach der Komplexität des Zählproblems (Wie viele Knoten befinden sich im Netz?) untersucht und das Zählproblem in Bezug zu dem Problem der Verbreitung von Informationen in einer gerichteten Variante von dynamischen Netzen gesetzt. Der dritte Teil beschäftigt sich schließlich mit der wiederholten Berechnung von Aggregationsfunktionen (z. B. das Maximum der Eingaben aller Knoten) in stabileren Varianten dynamischer Netze.

Reviewers:

- Prof. Dr. math. Friedhelm Meyer auf der Heide
  University of Paderborn, Germany
- Prof. Dr. rer. nat. Christian Scheideler
  University of Paderborn, Germany

To my parents.

Contents

List of Definitions and Theorems

List of Figures

List of Algorithms

Preface

First and foremost, I would like to thank my supervisor Friedhelm Meyer auf der Heide. He not only gave me the opportunity to start working on my doctorate directly after my Bachelor's degree but also supported me in many important decisions and key questions. Besides his extraordinary strength of comprehension, I admire his ability to reframe problems and see them from different angles.

I would also like to thank all current and former colleagues from the research group "Algorithms & Complexity" working with me at the Heinz Nixdorf Institute. It was a pleasure for me to conduct research with you. In addition, I would like to thank the team of the Paderborn Institute for Advanced Studies in Computer Science and Engineering (PACE) and the International Graduate School "Dynamic Intelligent Systems" that gave me a scholarship.

Finally, I would like to thank my parents, who have always been supportive and encouraged me relentlessly during my studies. Without the help of all these people I would not be where I am today and this thesis would certainly not exist.

—Sebastian Abshoff
February 2015
Paderborn, Germany

CHAPTER 1

Introduction

WIRELESS networks have to meet many challenges that are not present in wired networks. Some of them are caused by the characteristics of electromagnetic waves that are used for transmission. First of all, the strength of a wireless signal decreases with increasing distance to the sender, even in free space. This is known as the free-space path loss. Then, a wave tends to be refracted when it passes from one medium to another. When a wave hits a barrier, it is reflected and changes its direction. In addition, when a wave passes through a hole in a barrier or along an edge, it diffracts and changes its direction as well. Furthermore, waves can interfere, i.e., two waves can superpose and extinguish each other. Finally, there is an omnipresent electromagnetic background noise that also interferes with the sender's signal. These characteristics can lead to the effect that a signal—even if there is only one sender active at a time—cannot be observed by a receiver at all or can only be observed sometimes or infrequently.

There are many scenarios in which many devices create wireless networks to perform a certain task. For example, sensor networks consist of many cheap sensor nodes, which are deployed in an area to observe it and build wireless ad-hoc networks to transmit the observed data. Another example arises from smartphones that are able to communicate with close-by smartphones via technologies such as Bluetooth, WiFi, or Near Field Communication. In addition, these smartphones are equipped with more and more sensors nowadays, e.g., accelerometers, magnetometers, gyroscopic, light, temperature, pressure and humidity sensors to only name a few. In many of these scenarios, the devices can also be mobile or the environment in which they are operating might change or devices could fail over time. These considerations lead to the insight that, in general, large systems of many wireless devices are not static but very *dynamic*. From an algorithmic perspective, problems and questions arise in dynamic systems that are not present in static systems and thus techniques designed for static networks fail.

The onset of this work is a model by Fabian Kuhn, Nancy Lynch, and Rotem Oshman [KLO10] that studies wireless networks from a worst-case perspective. They model a wireless network as an edge-dynamic graph in a discrete, synchronous round model. Here, the wireless communication aspect is modeled in the communication model, in which a message of limited size sent by a node in an arbitrary round is delivered to all the node's neighbors in the succeeding round. The sequence of graphs is controlled by an adaptive adversary that tries to interfere with the execution of distributed algorithms. This adversary is allowed to change the network completely from round to round and its only restriction is that it must keep the network connected.

In this thesis, the influence of this kind of dynamic on the complexity of fundamental problems is analyzed. For the same purpose, Kuhn et al. [KLO10] introduce and study the k-token dissemination problem, in which k tokens—k pieces of information that are initially somewhere in the network—have to be disseminated to all nodes by a distributed algorithm. Compared to static networks containing n nodes, in which this problem can be solved in $O(n + k)$ rounds, they show on the one hand that it is possible to solve this problem in $O(nk)$ rounds and on the other hand that a naturally restricted class of algorithms cannot solve it in less than $\Omega(nk)$ rounds.

1.1 Thesis Overview

The main model adapted from [KLO10] and the notation are presented in Chapter 2 along with related work that is relevant for all parts of this thesis.

Subsequently, there is a chapter for each of the following three publications, in which the main model is refined and related work with respect to the corresponding chapter is reviewed.

Geometric Dynamic Networks (Chapter 3) The dynamic network model by Kuhn et al. [KLO10] is extreme in the sense that the adversary is allowed to change the network completely from one round to another. In this chapter, a restricted adversary model is introduced, in which nodes have geometric positions and nodes can be moved with some limited speed $v_{\max}$ only. Two nodes are connected if and only if their distance is below some fixed value. The main question that is answered in this chapter is whether it is possible to reproduce the existing lower bounds for general dynamic networks under these geometric restrictions. On the one hand, it turns out that for a restricted class of algorithms, a constant speed is already sufficient to give an $\Omega(nk)$ lower bound for the problem of disseminating k tokens in a network with n nodes. On the other hand, for maximum speed $v_{\max} \in o(1)$, it is shown that this problem can be solved in $\Theta(nk \cdot v_{\max})$ rounds, i.e., the smaller the maximum speed $v_{\max}$, the faster the problem can be solved. This chapter is based on the following publication.

> **2013** (with M. Benter, A. Cord-Landwehr, M. Malatyali, and F. Meyer auf der Heide). "Token Dissemination in Geometric Dynamic Networks." In: *Algorithms for Sensor Systems - 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, ALGOSENSORS 2013, Sophia Antipolis, France, September 5-6, 2013, Revised Selected Papers*, cf. [Abs+13a].

Counting versus Token Dissemination (Chapter 4) There are some results for information dissemination problems in dynamic networks, both upper and lower bounds. It is known that the problem of counting all nodes in a dynamic network can be solved by disseminating the unique IDs of all nodes in the network in $O(n^2)$ rounds. This is already the best known upper bound and it is not known whether it is possible to solve the counting problem faster. This chapter establishes a relation between the counting problem and the problem of disseminating *n* tokens from one node to another in a directed dynamic network of *n* nodes. While the complexity of this special token dissemination problem is not known, the construction given in this chapter could be used to derive a lower bound for counting. The results presented in this chapter are based on the following publication.

> **2013** (with M. Benter, M. Malatyali, and F. Meyer auf der Heide). "On Two-Party Communication through Dynamic Networks." In: *Principles of Distributed Systems - 17th International Conference, OPODIS 2013, Nice, France, December 16-18, 2013. Proceedings*, cf. [Abs+13b].

Continuous Aggregation in Dynamic Networks (Chapter 5) Beyond information dissemination and counting, this chapter introduces problems such as computing the maximum or the sum of the values measured by all nodes in a dynamic network. In addition, this chapter addresses continuous variants of these problems, in which all nodes have to compute the maximum or the sum of all values measured in several rounds. To compare algorithms solving these problems, two performance metrics, the *delay* and the *output rate*, are introduced. It turns out that there are better techniques to compute the continuous variants than just to execute an algorithm for the noncontinuous problem over and over again, i.e., there are algorithms that have a higher output rate and compute more results per time interval. This chapter is based on the following publication.

> **2014** (with F. Meyer auf der Heide). "Continuous Aggregation in Dynamic Ad-Hoc Networks." In: *Structural Information and Communication Complexity - 21st International Colloquium, SIROCCO 2014, Takayama, Japan, July 23-25, 2014. Proceedings*, cf. [AM14].

The thesis closes with a conclusion and discusses possible future research directions in Chapter 6.

CHAPTER 2

Preliminaries

THIS chapter introduces the notation and the dynamic network models used throughout this thesis. It discusses existing algorithms and bounds and gives an overview of related work.

2.1 Notation

- To avoid ambiguities, $\mathbb{N}_{\geq 0}$ and $\mathbb{N}^+$ are used to denote the natural numbers containing 0 and not containing 0, respectively.
- For $n \in \mathbb{N}^+$, $[n]$ is defined as the set $\{1, 2, \ldots, n\}$.
- For a set $A \subseteq \mathbb{N}^+$, $[A]$ denotes some set $[e]$ with $e \in A$.
- $\text{poly}(n) := \{f(n) \mid \exists \text{ constant } c \geq 0 \text{ such that } f(n) \in \mathrm{O}(n^c)\}$
- For a set S, $\mathcal{P}(S)$ denotes the set of all subsets of S.
- For an undirected graph $G = (V, E)$ and a node $v \in V$, $N(v)$ denotes the set of neighbors $\{u \in V \mid \{u, v\} \in E\}$. Accordingly, for a directed

graph $G = (V, E)$ and a node $v \in V$, $N^+(v)$ denotes the set of successors $\{u \in V \mid (v, u) \in E\}$ and $N^-(v)$ denotes the set of predecessors $\{u \in V \mid (u, v) \in E\}$.

- For an undirected graph $G = (V, E)$, G^D denotes the D^{th} power of G, i.e., the graph $G^D = (V, E^D)$ with

$$E^D = \{\{u, v\} \mid \exists \text{path between } u, v \in V \text{ of length} \leq D \text{ in } G\}\,.$$

- The term *with high probability* (w.h.p.) is used to refer to a probability of at least $1 - \mathrm{O}\left(\frac{1}{n^c}\right)$ for some constant $c > 0$.

2.2 The Dynamic Network Model

The following definitions have been adapted from Kuhn et al. [KLO10] and Oshman [Osh12] to unify the presentation in the subsequent chapters.

Dynamic Networks In short, a dynamic network is modeled as a time-varying graph with a fixed set of identifiable nodes in a discrete, synchronous time model.

Definition 2.1 ((Un)directed Dynamic Network)**.** An *(edge-)dynamic network* $\mathcal{G}$ is a triple (V, id, S) where V is a set of n nodes, $\mathrm{id} : V \to \mathcal{U}$ is an injective function with $\mathcal{U} \cong [\mathrm{poly}(n)]$, and $S = (G_r)_{r \in \mathbb{N}_{\geq 0}}$ is a sequence of undirected or directed graphs with $G_r = (V, E(r))$. For a node $v \in V$, $\mathrm{id}(v)$ is called *unique identifier (ID) of v* and $\mathcal{U}$ is called the *ID universe*. Furthermore, the parameter r is referred to as the *round* and G_r is called the *graph of round r*. In an *undirected* dynamic network, $E : \mathbb{N}_{\geq 0} \to \mathcal{P}(\{\{u, v\} \mid u, v \in V\})$ is a function mapping a round r to a set of undirected edges $E(r)$. Accordingly, in a *directed* dynamic network, $E : \mathbb{N}_{\geq 0} \to \mathcal{P}(V \times V)$ maps round r to a set of directed edges $E(r)$.

Based on this definition, a dynamic network is not even required to be connected. The following two definitions restrict the edge dynamic in a reasonable manner.

Definition 2.2 (T-Stability)**.** For $T \in \mathbb{N}^+$, a dynamic network is called *T-stable* if G_r is (strongly) connected in each round r and changes every T rounds only, i.e., $G_{iT} = G_{iT+1} = \ldots = G_{iT+(T-1)}$ for all $i \in \mathbb{N}_{\geq 0}$.

Definition 2.3 (T-Interval Connectivity)**.** For $T \in \mathbb{N}^+$, a dynamic network is called *T-interval connected* if the graph $G_{r,T} = \left(V, \bigcap_{i=r}^{r+(T-1)} E(i)\right)$ is (strongly) connected for all $r \in \mathbb{N}_{\geq 0}$.

Haeupler and Karger [HK11] assume T-stability while Kuhn et al. [KLO10] assume T-interval connectivity. Throughout this thesis, all dynamic networks are assumed to be at least (strongly) connected in each round, i.e., they are at least 1-stable and 1-interval connected. Furthermore, all results with respect to T-stability and T-interval connectivity are stated under the assumption that $T = \mathrm{O}(n)$.

Models of Communication and Computation Every node of a dynamic network is a computational entity. Starting in round 0, each node v is given its unique ID $\mathrm{id}(v)$. In general, the dynamic network itself is not known by the nodes but, depending on the assumptions, additional information about the dynamic network may be given to the nodes, e.g., the number of nodes n or T if the network is T-stable or T-interval connected. Furthermore, each node may be given some problem input. Based on this information, each node can perform computations and send and receive messages. In particular, per round r, each node

(1) receives messages from the preceding round $r - 1$,

(2) performs some computations based on available information and messages from the current and all previous rounds,

(3) can send a message, which is delivered in the next round $r + 1$.

To model the characteristics of wireless communication, the following communication model is used.

Definition 2.4 (Local Broadcast Communication)**.** A message sent by some node v in round r is delivered to all its neighbors $N(v)$ (successors $N^+(v)$) in the graph G_{r+1} of the following round.

In particular in 1-stable / 1-interval connected dynamic networks, a node sending a message is in general unaware of its neighbors (successors) in the following round. Thus, it can only assume that some other node will receive its message but it does not know which one.

While the computational power and memory of each node are assumed to be unbounded, the message size is limited, usually to a logarithmic number of bits which allows each node to send its unique ID (this is a widely used assumption in the distributed computing community, see, e.g., the CONGEST model by Peleg [Pel00]).

Beyond that, the distributed algorithms executed on the nodes of a dynamic network are required to terminate and to decide on an output, i.e., each node must reach a final state in finite time, in which it outputs the result of its computation and neither sends any message nor reacts to any incoming message. Then, the number of rounds an algorithm consumes to solve a problem instance in a dynamic network is defined as the number of rounds until all nodes have reached this final state.

Adversarial Network Dynamic The network dynamic is studied from a worst-case perspective, in which a distributed algorithm executed on the nodes of the dynamic network has to deal with an adversary that changes the network. The distributed algorithm and the adversary have different abilities and pursue opposing goals: the distributed algorithm chooses the messages sent by a node in order to terminate as soon as possible with the correct result, while the adversary defines the network in order to delay the termination of the algorithm as much as possible. The following three adversaries are relevant for this thesis. For deterministic algorithms, all three adversaries have the same strength, while for randomized algorithms, the power of the adversary can make a difference.

Definition 2.5 (Oblivious Adversary)**.** The *oblivious adversary* chooses the dynamic network before the execution of the algorithm starts.

Definition 2.6 (Weakly Adaptive Adversary)**.** The *weakly adaptive adversary* constructs the dynamic network during the execution of the algorithm based on its decisions. In particular, the graph of round $r + 1$ is defined before the execution of the algorithm in round r, i.e., the adversary is not aware of the outcome of coin flips made by a randomized algorithm in round r.

Definition 2.7 (Strongly Adaptive Adversary)**.** The *strongly adaptive adversary* also constructs the dynamic network during the execution of the algorithm based on its decisions. In particular, the graph of round $r + 1$ is defined after the execution of the algorithm in round r, i.e., the adversary is aware of the outcomes of coin flips made by a randomized algorithm in round r.

2.3 Fundamental Problems

There are three basic problems that have been studied in the model by Kuhn et al. [KLO10]. The first two formalize the spreading of information in the network. Here, a token refers to some piece of information that has to be disseminated to all nodes of the network.

Problem 2.8 (k-Token Dissemination)**.** As input $I(u)$, each node u is given a possibly empty subset from some known token universe $\mathcal{T} \cong [\text{poly}(n)]$ such that $\left|\bigcup_{v \in V} I(v)\right| = k$. The nodes of the network should disseminate these k tokens such that each node decides on $\bigcup_{v \in V} I(v)$. The parameter k is not known by the nodes beforehand.

Problem 2.9 (All-to-All Token Dissemination)**.** This problem is an instance of the k-token dissemination problem with $k = n$, in which each node is given exactly one token as input.

Another very fundamental problem is the counting problem. As defined in the model, the nodes are generally unaware of the number of nodes in the network. Note that an algorithm for the all-to-all token dissemination problem can be used to solve the counting problem by treating the nodes' unique IDs as tokens.

Problem 2.10 (Counting)**.** All nodes should decide on n, the number of nodes in the network.

2.4 Existing Results

This section describes some of the existing results in detail since they are used in Chapter 3 and Chapter 5 and motivate the question studied in Chapter 4. In particular, the token dissemination algorithm presented in Section 2.4.1 is applied to geometric dynamic networks in Chapter 3 and used to disseminate partial results of aggregation problems in Chapter 5. Furthermore, the lower bound presented in Section 2.4.2 is transferred to geometric dynamic networks in Chapter 3. Both lower bounds presented in Section 2.4.2 motivate the question of Chapter 4, whether the counting problem can be solved faster than all-to-all token dissemination. Finally, the randomized network coding based dissemination algorithm described in Section 2.4.3 is also used for disseminating partial results in Chapter 5.

2.4.1 The Algorithm by Kuhn, Lynch, and Oshman

This section describes deterministic algorithms by Kuhn et al. [KLO10] and Oshman [Osh12], which can be used in 1-stable / 1-interval connected dynamic networks to solve the counting problem in $O(n^2)$ rounds and the k-token dissemination problem in $O(n(n+k))$ rounds if n is unknown and in $O(nk)$ rounds if n is known beforehand. It is described in a top-down fashion and considers also T-stable or T-interval connected dynamic networks in which the stable (sub)graph is always C-vertex connected in addition.

Assuming the number of nodes is known or obtained with the algorithm by Kuhn et al., it is easy to see that one token can be disseminated to all nodes in $n-1$ rounds if each node sends this token up to round $n-2$ every round, starting in round 0 (if the token is given to the node as input) or in the round it has learned about the token (if it does not know about it from the beginning). To disseminate k tokens, one could dedicate the first $n-1$ rounds to the smallest token, the next $n-1$ rounds to the second smallest token, and so forth. In the i^{th} of these blocks of $n-1$ rounds, in each round, each node sends the i^{th} smallest token based on the set of tokens it knows or the empty message (referred to as $\perp$) otherwise. Thus, after $k(n-1)$ rounds all nodes know all tokens and in further $n-1$ rounds they all send the empty message. This observation can be used to derive a termination criterion, i.e., as soon as a node knows k' tokens and it does not learn about any new token from round $k'(n-1)$ to round $k'(n-1)+n-2$, it knows that $k'=k$ and it is safe to terminate.

If the dynamic network is T-stable or T-interval connected and T is known, the dissemination process can be sped up by a factor of T. Assume the dynamic network is $2T$-stable or $2T$-interval connected, i.e., there is a (sub)graph that does not change for $2T$ rounds. Then, it is possible to establish a pipelining effect such that in $2T$ rounds the T smallest tokens can be learned by T nodes. If this (sub)graph is C-vertex connected in addition, then the dissemination process can be sped up by another factor of C since the T smallest tokens can be learned by at least TC nodes. Listing 2.1 shows a pseudocode description of the algorithm for the k-token dissemination problem in T-stable or T-interval connected dynamic networks, in which the stable (sub)graph is C-vertex connected. This algorithm is used in Chapter 3 and in Chapter 5.

Algorithm 2.1: k-Token Dissemination under $2T$-Stability / $2T$-Interval Connectivity

```
A ← {input tokens} {* tokens the node is aware of *}
S ← ∅ {* tokens that are not sent anymore *}
r ← 0 {* current round number *}
repeat
  for i ← 1, ..., ⌈n/(TC)⌉ do
    P ← ∅ {* tokens already sent through pipeline *}
    for j ← 1, ..., 2T do
      t ← ⊥ {* token message that will be sent next *}
      if A ≠ S ∪ P then
        t ← smallest token in A \ (S ∪ P)
        P ← P ∪ {t}
      send t
      r ← r + 1
      A ← A ∪ {tokens received from neighbors}
  S ← S ∪ {T smallest tokens in A \ S}
until ⌈|A|/T⌉ · ⌈n/(TC)⌉ · 2T < r
return A
```

Theorem 2.11 (k-Token Dissemination under T-Stability / T-Interval Connectivity)**.** *In a T-stable or T-interval connected dynamic network, in which the stable (sub)graph is always C-vertex connected, the k-token dissemination problem can be solved by a deterministic algorithm in $\mathrm{O}\left(n + k + \frac{nk}{TC}\right)$ rounds if T and C and the number of nodes n in the network are known beforehand. [KLO10; Osh12]*

Remark 2.12 (k-Token Dissemination Having an Upper Bound on n). Note that this algorithm has asymptotically the same running time if a constant multiple of n is used instead of the exact count n.

To solve the counting problem, assume there is an algorithm that solves the following problem in $\mathrm{O}\left(k + \frac{k^2}{TC}\right)$ rounds in a T-stable or T-interval connected dynamic network, in which the stable (sub)graph is always C-vertex connected.

Problem 2.13 (k-Verification)**.** All nodes should decide whether $n \leq k$, i.e., whether a given parameter k is an upper bound on the number of nodes in the network.

Based on this algorithm, it is possible to determine the number of nodes. For this, an upper bound N on n can be found by running the k-verification algorithm for different values of k, starting with $k = 2^0, 2^1, 2^2, \ldots$ until $N = 2^{\lceil \log n \rceil} \geq n$. This requires at most

$$\sum_{i=0}^{\lceil \log n \rceil} \mathrm{O}\left(2^i + \frac{\left(2^i\right)^2}{TC}\right) = \mathrm{O}\left(n + \frac{n^2}{TC}\right)$$

rounds. Since $n \leq N \leq 2n$, the token dissemination algorithm could be used to disseminate all unique IDs such that all nodes could obtain the exact count n after further $\mathrm{O}\left(n + \frac{n^2}{TC}\right)$ rounds. However, this is not necessary since the method previously used to solve the last k-verification problem implicitly disseminates all unique IDs to all nodes.

Theorem 2.14 (Counting under T-Stability / T-Interval Connectivity)**.** *In a T-stable or T-interval connected dynamic network, in which the stable (sub)graph is always C-vertex connected, the counting problem can be solved by a deterministic algorithm in $\mathrm{O}\left(n + \frac{n^2}{TC}\right)$ rounds if T and C are known beforehand. [KLO10; Osh12]*

To solve the k-verification problem, assume there is yet another algorithm that solves the following problem in $\mathrm{O}\left(k + \frac{k^2}{TC}\right)$ rounds.

Problem 2.15 (k-Committee Election)**.** All nodes should build *committees* of size $\leq k$, i.e., each node v should decide on a *committee ID c_v* from some universe $\mathcal{C} \cong [\mathrm{poly}(n)]$ such that $|\{v \in V : c_v = c\}| \leq k$ for each $c \in \mathcal{C}$. If $k \geq n$, then there must be at most one committee, i.e., all nodes must decide on the same committee ID c.

Having computed a solution for the k-committee election problem, it is easy to solve the k-verification problem: all nodes check whether there is only one committee. This can be done if each node v starts sending its committee ID c_v. In further $\left\lceil \frac{k}{C} \right\rceil - 1$ rounds, each node either continues sending c_v if it has not yet received anything different from its own committee ID or it sends the empty message (assuming $\perp \notin \mathcal{C}$). If there is more than one committee, then at least C nodes sending the same committee ID must be connected to nodes not sending this committee ID. Thus, after $\mathrm{O}\left(\frac{k}{C}\right)$ rounds, if and only if each node has received something different from its own committee, then there is more than one committee.

Finally, it is left to show how the k-committee election problem can be solved. In the first part of the following algorithm, one or more leaders are elected by propagating the smallest unique ID for $k-1$ rounds. Each node that has not received a smaller unique ID considers itself a leader. Note that while $k < n$, there might be more than one leader, but there is exactly one leader if $k \geq n$. In the second part, these leaders invite up to $k-1$ other nodes to join their committee. For this, the following phases are executed in $k-1$ cycles:

(1) *Polling Phase*: For $k-1$ rounds, the nodes propagate the smallest unique ID they have received which has not yet joined a committee.

(2) *Invitation Phase*: A leader selects the smallest unique ID it has received in the polling phase and sends an invitation message containing this unique ID and its own unique ID. This invitation is propagated for $k-1$ rounds. Upon reception of an invitation message, the invited node joins the committee of the leader, i.e., its committee ID is the one of the inviting leader. If a node is invited to more than one committee, it either joins only one or none of the committees.

If subsequently a node has not yet joined a committee, it joins its own committee.

This algorithm solves the k-committee election problem in $\mathrm{O}(k^2)$ rounds, since each committee consists of at most k nodes and if $k \geq n$, then there is exactly one leader and hence there is exactly one committee. As in the k-token dissemination problem, this algorithm can be sped up in T-stable or T-interval connected dynamic networks, in which the stable (sub)graph is always C-vertex connected: In the first part, the leaders can be elected in $\Theta\left(\frac{k}{C}\right)$ rounds. In the second part, the polling phase and invitation phase, $\Theta(T)$ unique IDs and $\Theta(T)$ invitations can be propagated in $\Theta\left(\frac{k}{C} + T\right)$ rounds.

Lemma 2.16 (k-Committee Election under T-Stability / T-Interval Connectivity). *In a T-stable or T-interval connected dynamic network, in which the stable (sub)graph is always C-vertex connected, the k-committee election problem can be solved by a deterministic algorithm in $\mathrm{O}\left(k + \frac{k^2}{TC}\right)$ rounds if T and C are known beforehand. [KLO10; Osh12]*

2.4.2 Lower Bounds for Token-Forwarding Algorithms

There are lower bounds for algorithms that only store and forward tokens based on the following definition. Intuitively, it prevents the algorithm from using coding techniques and combining multiple tokens into one message.

Definition 2.17 (Token-Forwarding Algorithm). Let $A_v(r)$ denote the set of tokens node v has received by or at the beginning of round r including its input $I(v)$. A *token-forwarding algorithm* is allowed to send only one token from $A_v(r)$ or the empty message ($\perp$), i.e., it must either send a single token without modification and without annotation or the empty message. Furthermore, it must not terminate before $A_v(r) = \left|\bigcup_{v \in V} I(v)\right|$.

Lower Bound for Knowledge-Based Token-Forwarding Algorithms

The following definition restricts the abilities of the algorithm to choose a token. For this class of algorithms, Kuhn et al. [KLO10] and Oshman [Osh12] prove an $\Omega\left(n + \frac{nk}{T}\right)$ lower bound for the k-token dissemination problem. This technique can be transferred to a geometric version of dynamic networks as shown in Chapter 3.

Definition 2.18 (Knowledge-Based Token-Forwarding Algorithm). A token-forwarding algorithm is called *knowledge-based*[1] if the probability distribution that determines which token is sent by node v in round r is only a function of

(1) its unique ID $\mathrm{id}(v)$,

(2) $A_v(0), \dots, A_v(r)$,

(3) the sequence of v's coin tosses up to round r (including r).

[1] Oshman [Osh12] introduced a similar definition to the one used here and in [KLO10]. They refer to this class of algorithms as *limited-history algorithms*.

Most importantly, this definition allows for finding infrequently sent tokens regardless of the dynamic network's evolution (for details, see Chapter 3). The following result is known for T-stable dynamic networks and T-interval connected dynamic networks.

Theorem 2.19 (Lower Bound for Knowledge-Based Token-Forwarding Algorithms)**.** *In a T-stable or T-interval connected dynamic network controlled by a weakly adaptive adversary, any knowledge-based token-forwarding algorithm for k-token dissemination requires $\Omega\left(n + \frac{nk}{T}\right)$ rounds to succeed with probability $> \frac{1}{2}$. [KLO10; Osh12]*

Lower Bound for Centralized Token-Forwarding Algorithms

A lower bound for centralized token-forwarding algorithms under a strongly adaptive adversary is introduced by Dutta et al. [Dut+13]. Haeupler and Kuhn [HK12] extend this technique for several problem variations.

To prove an $\Omega\left(\frac{nk}{\log n}\right)$ lower bound for 1-stable / 1-interval connected dynamic networks, consider the potential function

$$\Phi(r) := \sum_{u \in V} |A_u(r) \cup A'_u| .$$

Here, as defined previously, $A_u(r)$ is the set of tokens node u has received by or at the beginning of round r including its input $I(u)$. For each node $u \in V$, $A'_u \subseteq \mathcal{T}$ is a subset of tokens that is used to define a greedy adversary that, depending on the choices made by the token-forwarding algorithm, chooses the next graph. Haeupler and Kuhn [HK12] show that there is a way to define the sets A'_u such that

(1) $\sum_{u \in V} |A'_u| < 0.3nk$ and

(2) an adversary is able to greedily choose a graph such that the potential is increased by $\mathrm{O}(\log n)$ per round.

Thus, if $\sum_{u \in V} |A_u(0)| \leq \frac{1}{2}nk$, i.e., each node receives $\leq \frac{k}{2}$ tokens as input on average, then the $\Omega\left(\frac{nk}{\log n}\right)$ lower bound follows immediately.

To show (1) and (2), Haeupler and Kuhn [HK12] use the probabilistic method: Note that an edge $\{u, v\}$ does not increase the potential if node u sends a token in A'_v and node v sends a token in A'_u or if both nodes u and v send the same token. These edges are called *free* edges. If each set A'_u for each node u contains each token independently with probability $p = \frac{1}{4}$, then it is possible to prove that the graph induced by all free edges has $O(\log n)$ connected components with probability $\geq \frac{3}{4}$. Thus, there is a way to define the sets A'_u such that a greedy adversary can always choose all free edges and needs to add $O(\log n)$ edges to connect the remaining connected components.

In particular, if there are $\geq s$ connected components in the graph induced by all free edges, then there is a set S with $|S| = s$ such that all edges between nodes in S are not free. Each of these edges is not free with probability $1 - p^2$ (two nodes $u, v \in S$ must have been assigned different tokens, since otherwise the edge would have been free; the token sent by node u is in A'_v with probability p and the token sent by node v is in A'_u with probability p). For a fixed assignment of tokens to the nodes in S, the probability that all these edges are not free is $\leq (1 - p^2)^{\binom{s}{2}} < e^{-p^2 s^2/4}$. The probability that this is the case for all $\binom{n}{s}$ possibilities to choose s nodes and $< k^s$ possibilities to assign a token to each node in S is $< \binom{n}{s} k^s e^{-p^2 s^2/4} < \frac{1}{4}$ for $s = 12p^{-2}\ln(nk)$. Thus, the probability of the complementary event is $\geq \frac{3}{4}$.

A more sophisticated argument for T-interval connected dynamic networks yields the following result.

Theorem 2.20 (Lower Bound for Centralized Token-Forwarding Algorithms). *In a T-interval connected dynamic network controlled by a strongly adaptive adversary, any centralized token-forwarding algorithm for k-token dissemination requires* $\Omega\left(\frac{nk}{T^2 \log n}\right)$ *rounds. [HK12]*

2.4.3 Randomized Network Coding Algorithms

If the messages are allowed to contain not only a token but any $O(\log n)$-bit string, then the randomized network coding techniques by Haeupler and Karger [HK11] allow for faster token dissemination than $O(nk)$ rounds.

Random Linear Network Coding

Assuming the k tokens have a distinct indices in $[k]$, they can be disseminated as follows. Consider $k' = O(\log n)$ indexed tokens $t_1, \dots, t_{k'} \in \mathbb{F}_2^{O(\log n)}$ and build vectors v_i by concatenating t_i with the i^{th} basis vector e_i of $\mathbb{F}_2^{k'}$. In each round, each node sends a random linear combination of all vectors received so far (including the vectors $v_1, \dots, v_{k'}$). Then, after $O(n)$ rounds, all nodes are able to reconstruct $v_1, \dots, v_{k'}$ using Gaussian elimination with high probability. Applying this $\frac{k}{k'}$ times, k indexed tokens can be disseminated in $O\left(\frac{nk}{\log n}\right)$ rounds with high probability.

Token Indexing

Unfortunately, the k tokens are not indexed at the beginning. Therefore, Haeupler and Karger [HK11] show how to gather tokens at some node in the network such that they can be assigned indices locally on this node. They introduce two routines, which are outlined in the following.

- random-forward: It can be shown that if each node sends a random token for $O(n)$ rounds, then there is a node with at least $\sqrt{k}$ tokens with high probability. This node with the maximum token count is identifiable in $O(n)$ rounds.
- greedy-forward: By repeatedly executing the greedy-forward routine, the identifiable node with the maximum token count is able to index $O(\log n)$ tokens and then these tokens can be disseminated using the random linear network coding approach. This requires $O\left(\frac{nk}{\log n} + n \log n\right)$ rounds to solve the k-token dissemination problem with high probability.

More sophisticated forwarding and dissemination routines for T-stable dynamic networks yield the following result. These routines include the graph patching technique used in Chapter 5, which involves the computation of maximal independent sets. For different ranges of T, different routines become efficient. This is why there is a minimum in the running time.

Theorem 2.21 (k-Token Dissemination with Network Coding)**.** *In a T-stable dynamic network controlled by a weakly adaptive adversary, the k-token dissemination problem can be solved by a randomized network coding algorithm in*

$$O\left(\min\left\{\frac{nk}{T^2} + T^2 n \log^2 n, \frac{nk \log n}{T^2} + Tn \log^2 n, \frac{n^2 \log n}{T^2} + n \log n\right\}\right)$$

rounds with high probability. [HK11]

2.5 Related Work

There is a huge amount of related work on network dynamics and how to model them. For example, in the research field of self-stabilization, starting with the work by Dijkstra [Dij74], a system should provably converge to a desirable configuration from any initial configuration assuming there are no further changes. In other works, dynamic is modeled by never ending random processes, which change the state of a system. For more information, see, e.g., the survey by Kuhn and Oshman [KO11a] on random (and adversarial) dynamic network models. The related work discussed here and in the following chapters is mainly limited to adversarial network dynamics similar to those described in Section 2.2.

Casteigts et al. [Cas+12] introduce a framework called time-varying graphs (TVGs) that unifies several models in different research areas of dynamic systems. The dynamic network models used in this thesis can also be classified according to this framework.

Brandes and Meyer auf der Heide [BM12] study counting problems in dynamic networks, in which, in addition to the dynamic network model introduced in Section 2.2, the transmission of a message may fail with probability p. On the one hand, they show that strong counting with known p is impossible. Here, *strong counting* refers to algorithms that stop with the correct count within a runtime bound $t(n)$. On the other hand, they show that weak counting, which does not require termination, is possible and that strong counting becomes possible if an upper bound on n is given.

O'Dell and Wattenhofer [OW05] analyze information dissemination problems in slightly different but worst-case adversarial models and evaluate the requirements necessary for disseminating information. More specifically, these requirements include correctness, termination, storage at each node, and uniformity (i.e., the availability of unique IDs and the node count).

Augustine et al. [Aug+12] consider a dynamic network model that also takes node churn into account, i.e., nodes join and leave the network. In particular, they assume that the network size does not change, but a fraction of the nodes is replaced in each round. Furthermore, they assume—instead requiring connectivity only—that the graph of each round is a vertex expander. In this model, they study agreement problems and show that there is a randomized algorithm that achieves almost-everywhere agreement in $O\left(\log^2 n\right)$ rounds with high probability under an oblivious adversary and up to linear churn per round. Under an adaptive adversary and churn up to $\epsilon\sqrt{n}$ per round, they achieve almost-everywhere agreement in $O\left(\log m \log^3 n\right)$ rounds with high probability, where m is the size of the input value domain.

Michail et al. [MCS12b; MCS14] relax the requirement that the adversary must keep the dynamic network connected in each round. They introduce three temporal connectivity metrics and study their relation to other metrics such as the dynamic diameter: the *outgoing influence time*, which is the maximal time until the state of a node influences the state of another node, the *incoming influence time*, which is the maximal time until the state of a node is influenced by the state of another node, and the *connectivity time*, which is the maximal time until two parts from a cut of the network become connected.

Kuhn et al. [KMO11] study consensus problems in dynamic networks, in which all nodes have to agree on the input of one node. In contrast to the dynamic network model used in this thesis, the size of a message is not limited. Consensus problems are also considered by Biely et al. [BRS12] in directed dynamic networks that are not strongly connected in each round. Kuhn et al. [Kuh+10] study clock synchronization problems in a different time model.

CHAPTER 3

Geometric Dynamic Networks

POWERFUL adversaries such as those in the model by Kuhn et al. [KLO10] slow down the dissemination of information in the network: Compared to static networks, in which the k-token dissemination problem can be solved by a simple algorithm in $O(n + k)$ rounds, $\Omega(nk)$ rounds is the best a knowledge-based token-forwarding algorithm can achieve in a 1-stable / 1-interval connected dynamic network. Even stronger token-forwarding algorithms require $\Omega\left(\frac{nk}{\log n}\right)$ rounds [Dut+13; HK12]. A question that arises is whether the assumption that the topology of the network could change arbitrarily from round to round is realistic. Indeed, it is reasonable to assume that nodes in a mobile ad-hoc network move with limited speed within a certain area and thus the dynamic is restricted by the geometry of the network.

These thoughts lead to the model discussed in this chapter. In this model, each node has a variable position that is controlled by the adversary. The adversary cannot change these positions arbitrarily from round to round, but, as an effect of a speed limit $v_{\max}$ it has to adhere to, new positions depend on old positions. It is assumed that two nodes can communicate if their distance is smaller than some parameter R, which models the transmission range of their antennas. This defines a dynamic unit disc graph (UDG) with respect to radius R. To ensure connectivity, the adversary must keep the UDG connected with respect to radius 1.

Although the UDG model is way too simplistic to cover effects like wave refraction, diffraction, reflection, and interference that affect wireless communication in reality, it is sufficient enough to show an $\Omega(nk)$ lower bound for knowledge-based token-forwarding algorithms if $v_{\max}$ and R are constant and independent of n and k. Still, for smaller speed limits, there is an $O(n + k + nk \cdot v_{\max})$ upper bound and a matching $\Omega(n + k + nk \cdot v_{\max})$ lower bound for knowledge-based token-forwarding algorithms as shown in this chapter.

Chapter Basis The results presented in this chapter are based on the following publication.

> **2013** (with M. Benter, A. Cord-Landwehr, M. Malatyali, and F. Meyer auf der Heide). "Token Dissemination in Geometric Dynamic Networks." In: *Algorithms for Sensor Systems - 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, ALGOSENSORS 2013, Sophia Antipolis, France, September 5-6, 2013, Revised Selected Papers*, cf. [Abs+13a].

Chapter Outline First of all, the geometric dynamic network model is introduced and formally defined in Section 3.1. Then, literature related to geometric dynamic models is discussed in Section 3.2. In Section 3.3, properties of the geometric dynamic network model are proven. These yield an improved running time of the algorithm presented in Section 2.4.1. Section 3.4 gives a lower bound for knowledge-based token-forwarding algorithms that almost matches the upper bound. Finally, Section 3.5 concludes this chapter and discusses unanswered questions.

3.1 The Geometric Dynamic Network Model

The dynamic network model from Section 2.2 is changed in the following way. In each round r, each node v has a position $p_r(v) \in \mathbb{R}^2$ in the Euclidean plane.[1] The distance between two nodes u, v in round r is denoted by $d_r(u,v) := \|p_r(u) - p_r(v)\|_2$. For a node v and a set of nodes U, $d_r(v,U) := \min_{u \in U}\{d_r(u,v)\}$. Some fixed parameter $R \geq 1$ models the *communication range*, which is assumed to be the same for all nodes. The graph $G_r = (V, E(r))$ of round r is given by $E(r) := \{\{u,v\} \mid u,v \in V \wedge d_r(u,v) \leq R\}$, i.e., two nodes u, v are connected if and only if their distance is smaller than or equal to the communication range R. G_r is also referred to as the *communication graph* of round r. For technical reasons, it is assumed that $R = O(n)$.

In contrast to the original model, the adversaries in this chapter are no longer able to control the edges of the dynamic network directly. Instead, they are able to change the positions of the nodes under the following restrictions. A parameter $v_{\max} \geq 0$ models a *speed limit* the adversaries have to adhere for each node: The maximal distance each node can be moved per round is $v_{\max}$, i.e., $\|p_r(v) - p_{r+1}(v)\|_2 \leq v_{\max}$ for each round r and for each node v. Furthermore, the adversaries must keep the *connectivity graph* $G'_r = (V, E'(r))$ with $E'(r) := \{\{u,v\} \mid u,v \in V \wedge d_r(u,v) \leq 1\}$ connected in each round r. The initial positions of the nodes in round 0 and the assignment of the unique IDs to the nodes are still under the control of the adversaries.

3.2 Related Work

Dynamic models, in which nodes are placed in a metric space and move with a bounded speed, have been studied, e.g., by Bienkowski et al. [Bie+09] for the page migration problem, which is a classical online problem in the field of data management. Here, the distance between two nodes models communication costs. A so-called *page* of size D is stored at one node $P(t)$ of the network. In each time step t, a request σ_t arrives at one node and causes costs of $d(\sigma_t, P(t)) + 1$. Then, the page can be moved from $P(t)$ to another node $P(t+1)$ causing costs of $D \cdot (d(P(t), P(t+1)) + 1)$. The authors present an $O(\min\{n \cdot \sqrt{D}, D\})$-competitive deterministic algorithm and a lower bound of $\Omega(\min\{n \cdot \sqrt{D}, D\})$ for randomized algorithms against adaptive-online adversaries. Furthermore, for randomized algorithms, they show that the competitive ratio against an oblivious adversary is $\Theta(\min\{\sqrt{D \cdot \log n}, D\})$.

[1] All results shown in this chapter also hold for any fixed dimension higher than 2.

Similar models are often (implicitly or explicitly) used in the context of local strategies for robotic formation problems (for a survey, see [KM11]). For example, Degener et al. [Deg+11] study the problem of gathering n robots in the Euclidean plane at one point in $O(n^2)$ rounds. Their local strategy does not require any communication, but the robots need to be able to sense their neighbors' positions. Given these positions, each robot moves to a point within constant distance and thus all robots move with limited speed. Furthermore, their strategy guarantees that an initially connected UDG with respect to radius 1 remains connected during the gathering process. In related scenarios, in which connections are also removed but the UDG is left connected, the token dissemination algorithm described in this chapter can be applied during gathering to exchange information (e.g., sensor data) on top of a unit disc communication graph.

The UDG model has been studied extensively in the area of geographic routing algorithms for wireless ad-hoc and sensor networks. Geographic routing assumes nodes are able to obtain their own position from a location service (e.g., Galileo, GPS or GLONASS) and utilize this position information to deliver a packet from a source to a destination node. These algorithms focus on the delivery of single packets and do not consider the congestion arising in token dissemination problems. Kuhn et al. [KWZ03; Kuh+03] propose a geographic routing algorithm that is optimal in the worst-case and efficient in the average-case. Frey et al. [FRS09] give a broad overview about further geographic routing algorithms.

3.3 Upper Bounds for Token-Forwarding Algorithms

The geometric dynamic network model has some characteristics that can be exploited to speed up token dissemination for some ranges of R and $v_{\max}$.

Lemma 3.1 (Interval Connectivity in Geometric Dynamic Networks)**.** *A geometric dynamic network with communication range $R \geq 1$ and maximum speed $v_{\max}$ is $\left\lfloor \frac{R-1}{2v_{\max}} \right\rfloor + 1$-interval connected.*

Proof. Consider the connectivity graph $H \subseteq G_r$, i.e., the UDG with respect to radius 1. Note that H is a spanning subgraph of G_r and stable for further

$\left\lfloor \frac{R-1}{2v_{\max}} \right\rfloor$ rounds since any two nodes u and v that are connected in H have to cover a distance of at least $R-1$ to become disconnected. □

Lemma 3.2 (Stable Vertex Connected Subgraphs in Geometric Dynamic Networks)**.** *A geometric dynamic network with communication range $R \geq 2$ and maximum speed $v_{\max}$ contains a spanning $\left\lfloor \frac{1}{2}R \right\rfloor$-vertex connected subgraph that is stable for $\left\lfloor \frac{R}{4v_{\max}} \right\rfloor + 1$ rounds.*

Proof. Consider a subgraph $H \subseteq G_r$ defined as the UDG with respect to radius $\frac{1}{2}R$ in an arbitrary round r. Note that H is stable for further $\left\lfloor \frac{\frac{1}{2}R}{2v_{\max}} \right\rfloor$ rounds since any two nodes u and v that are connected in H have to cover a distance of at least $\frac{1}{2}R$ to become disconnected. Furthermore, note that, since the connectivity graph must be connected, the deletion of a set C of $\left\lfloor \frac{1}{2}R \right\rfloor - 1$ nodes cannot separate H into two connected components $A, B \neq \emptyset$ because

$$\underbrace{\left|\left\{u \in V \setminus A \mid d(u,A) \leq \frac{1}{2}R\right\}\right|}_{\#\text{nodes} \notin A \text{ reachable from nodes} \in A} \geq \left\lfloor \frac{1}{2}R \right\rfloor > |C| = \left\lfloor \frac{1}{2}R \right\rfloor - 1,$$

i.e., the nodes in A can reach at least one node that is not in $A \cup C$. Thus, H is a $\left\lfloor \frac{1}{2}R \right\rfloor$-vertex connected spanning subgraph that is stable for $\left\lfloor \frac{R}{4v_{\max}} \right\rfloor + 1$ rounds. □

Theorem 3.3 (k-Token Dissemination in Geometric Dynamic Networks)**.** *In geometric dynamic networks with communication range $R > 1$ and maximum speed $v_{\max}$, the k-token dissemination problem can be solved by a deterministic algorithm in*

$$\mathrm{O}\big(n + k + nk \cdot \min\{v_{\max}, R\} \cdot R^{-2}\big)$$

rounds if n, R, and $v_{\max}$ are known beforehand.

Proof. If $R > 1$, then according to Lemma 3.1 the geometric dynamic network is $\Omega\left(\max\left\{\frac{R}{v_{\max}}, 1\right\}\right)$-interval connected. Thus, by Theorem 2.11, the k-token dissemination problem can be solved by a deterministic algorithm in $\mathrm{O}\big(n + k + nk \cdot \min\{v_{\max}, R\} \cdot R^{-1}\big)$ rounds if n, R, and $v_{\max}$ are known beforehand.

If in addition $R \geq 2$, then by Lemma 3.2 the communication graph contains a spanning $\Theta(R)$-vertex connected subgraph that is stable for $\Omega\left(\max\left\{\frac{R}{v_{\max}}, 1\right\}\right)$

rounds. Thus, by Theorem 2.11, the k-token dissemination problem can be solved by a deterministic algorithm in $O(n + k + nk \cdot \min\{v_{\max}, R\} \cdot R^{-2})$ rounds if n, R, and $v_{\max}$ are known beforehand. □

Corollary 3.4 (Counting in Geometric Dynamic Networks)**.** *In geometric dynamic networks with communication range $R > 1$ and maximum speed $v_{\max}$, the counting problem can be solved by a deterministic algorithm in*

$$O(n + n^2 \cdot \min\{v_{\max}, R\} \cdot R^{-2})$$

rounds if R and $v_{\max}$ are known beforehand.

3.4 Lower Bounds for Token-Forwarding Algorithms

In this section, lower bounds for knowledge-based token-forwarding algorithms are presented. The analysis follows the one given by Kuhn et al. [KLO10] but takes the geometric restrictions of the adversary into account. To simplify the presentation, the proof is first given for communication range $R = 1$ and then it is generalized to $R \geq 1$.

Theorem 3.5 (Lower Bound for Knowledge-Based Token-Forwarding Algorithms in Geometric Dynamic Networks with $R = 1$)**.** *In geometric dynamic networks with communication range $R = 1$ and maximum speed $v_{\max}$, which are controlled by a weakly adaptive adversary, any knowledge-based token-forwarding algorithm requires*

$$\Omega(n + k + nk \cdot \min\{v_{\max}, 1\})$$

rounds to solve the k-token dissemination problem with probability $> \frac{1}{2}$.

Proof. The initial positions are assigned as shown in Figure 3.1. Let $\epsilon > 0$ suitably small, $v_{\max} = O(1)$, $L := \left\lceil \frac{1+\varepsilon}{2v_{max}} \right\rceil$ and $n - 4$ a multiple of L. Except for four nodes v_0, v_{n-3}, v_{n-2}, and v_{n-1}, all nodes are set up to horizontal lines of L nodes. The distance of a node to its neighbors on the same line is exactly 1. The lines are numbered from 0 to $\frac{n-4}{L} - 1$. Initially, all lines except for line 0 have the same horizontal and vertical positions such that multiple nodes share the same position. These lines are said to be at level 0. Line 0 is in distance $1 + \epsilon$ below all other lines in vertical direction and said to be at level 1. Level 2 is in distance

$1 + \epsilon$ below level 1 and level 3 is in distance $1 + \epsilon$ below level 2 in vertical direction. Both levels 2 and 3 are not occupied by any line of nodes at the beginning. Node v_0 is positioned in distance 1 to the left of the leftmost node of line 0 at level 1 such that—assuming level 2 was occupied—the distance to the leftmost node on level 2 is smaller than 1. Node v_{n-1} is positioned below v_0 such that the distance to v_0 is smaller than 1 and—assuming level 3 was occupied—the distance to the leftmost node on level 3 is exactly 1. Similarly, node v_{n-2} is positioned in distance 1 right to the rightmost node at level 0 such that the distance to the rightmost node at level 1 is smaller than 1. Finally, node v_{n-3} is positioned below v_{n-2} in a way, that its distance to almost all points below the rightmost node of level 1 in vertical direction and in distance at most 1 is smaller than 1 but—assuming level 2 was occupied—the distance of v_{n-2} to the rightmost node of level 2 is strictly greater than 1.

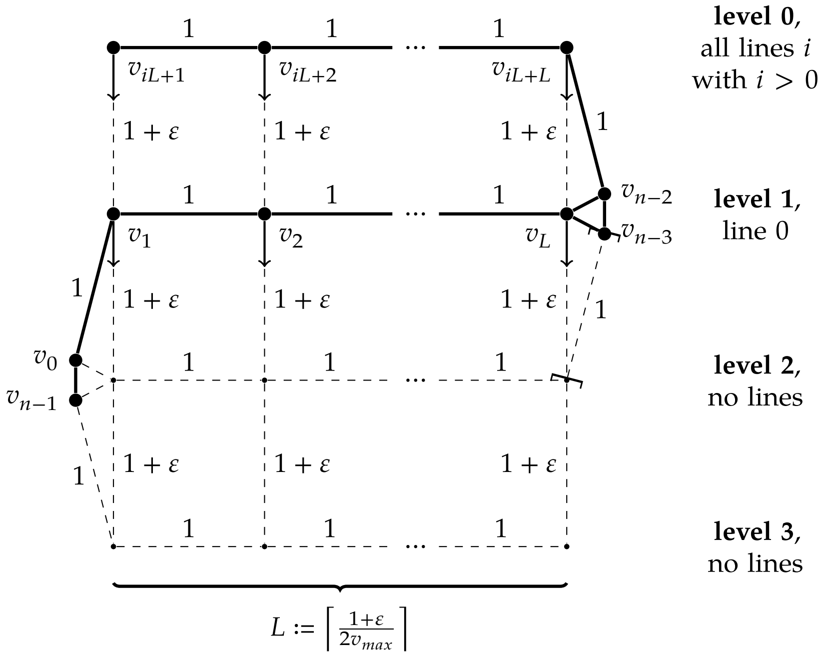

Figure 3.1: Lower bound construction for $R = 1$. Initial positions with multiple lines at level 0 and one line at level 1.

Now, v_0 is the node to which all k tokens are assigned. All other nodes do not know any token in the beginning. The key insight on knowledge-based token-forwarding algorithms is that the tokens' probability distribution used by node

v_0 does not depend on the evolution of the dynamic network since the set of tokens $A_{v_0}(r)$ is fixed for any $r \geq 0$ (cf. Definition 2.18). Thus, if the algorithm terminates by round $r^* = \left\lfloor \frac{(n-4)k}{2L} \right\rfloor - 1$, then by linearity of expectation and Markov's inequality, there must exist some infrequently sent token t which is sent $< \frac{n-4}{L}$ times by node v_0 with probability $> \frac{1}{2}$. Subsequently, it is shown that a geometrically restricted adversary can move the nodes in the plane such that the knowledge-based token-forwarding algorithm cannot disseminate token t to all nodes.

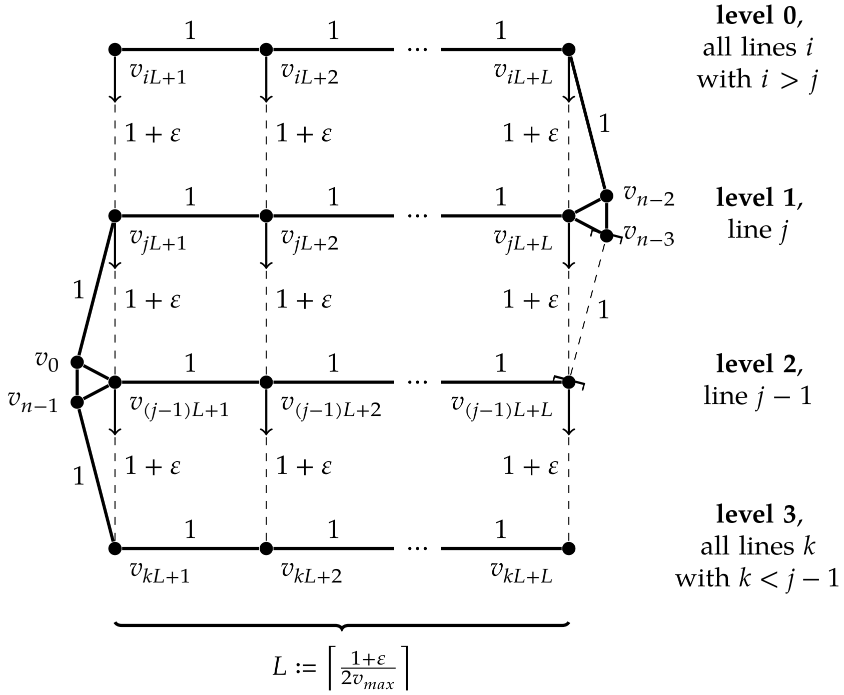

Figure 3.2: Lower bound construction for $R = 1$. Positions of all nodes after the infrequently sent token was sent for the $j - 1$ time.

When node v_0 sends token t for the first time, the adversary starts moving down lines 0 and 1 until they reach levels 1 and 2, respectively. Since the lines can be moved with maximum relative speed $2v_{\max}$, the levels can be reached in $\left\lceil \frac{1+\epsilon}{2v_{\max}} \right\rceil$ rounds. By choice of L, token t cannot reach node v_{n-3}. Thus, node v_0 is again the only node knowing token t that is connected to nodes at levels 0 and 1. When node v_0 sends token t for the second time, lines 2, 1, and 0 are moved to levels 1, 2, and 3, respectively. Again, by choice of L, token t cannot reach node v_{n-3}. In general, when token t is sent for the $(j-1)^{\text{th}}$ time, then lines $j+1$, j, and $j-1$ are moved to levels 1, 2, and 3, respectively. The situation before token t is sent for the $j-1^{\text{th}}$ time is depicted in Figure 3.2.

Note that the connectivity graph is connected in each round. Because node v_0 sends token t less than $\frac{n-4}{L}$ times, it cannot reach node v_{n-3} and thus the algorithm cannot be finished by round $r^* = \Omega(nk \cdot v_{\max})$ which concludes the proof of the theorem.[2] □

Theorem 3.6 (Lower Bound for Knowledge-Based Token-Forwarding Algorithms in Geometric Dynamic Networks with $R \geq 1$)**.** *In geometric dynamic networks with communication range $R \geq 1$ and maximum speed $v_{\max}$, which are controlled by a weakly adaptive adversary, any knowledge-based token-forwarding algorithm requires*

$$\Omega\big(n + k + nk \cdot \min\{v_{\max}, R\} \cdot R^{-3}\big)$$

rounds to solve the k-token dissemination problem with probability $> \frac{1}{2}$.

Proof. The initial positions are assigned in a similar way as before (cf. Figure 3.3). Let $\epsilon > 0$ be suitably small, $v_{\max} = O(1)$, $L := \left\lceil \frac{3(R+\varepsilon)}{2v_{max}} \right\rceil \cdot (R+1)$ and $n - 2\lceil 3(R+\epsilon) \rceil + 2$ a multiple of L. Except for $2\lceil 3(R+\epsilon) \rceil - 2$ nodes, namely v_0 and $v_{n-1}, \dots, v_{n-2\lceil 3(R+\epsilon)+3 \rceil}$, all nodes are set up to horizontal lines of L nodes. Now, there are 6 levels numbered from 0 to 5. Initially, all lines except for lines 0 and 1 are on level 0, and the lines 0 and 1 are on levels 1 and 2, respectively. The remaining nodes are used to connect the levels: $v_0, v_{n-1}, \dots, v_{n-\lceil 3(R+\epsilon) \rceil+3}$ connect level 2 down to level 5, and $v_{n-\lceil 3(R+\epsilon)+2 \rceil}, \dots, v_{n-2\lceil 3(R+\epsilon) \rceil+3}$ connect level 0 down to level 2 and all positions above level 3, i.e., the distance of $v_{n-2\lceil 3(R+\epsilon) \rceil+3}$ to almost all points below the rightmost node of level 2 in vertical direction in distance at most R is smaller than 1 but—assuming level 2 was occupied—the distance of $v_{n-2\lceil 3(R+\epsilon) \rceil+3}$ to rightmost node of level 3 is strictly greater than 1.

As before, the idea is to find a token that is sent infrequently over some cut. Yet, there is more than one node in the cut because G_r is $\lfloor R \rfloor$-vertex connected. For this reason, the nodes $v_0, v_{n-1}, \dots, v_{n-\lfloor R \rfloor+3}$ initially receive all k tokens

[2]Note that even in a static network $\Omega(n+k)$ rounds are required if $\Theta(k)$ tokens are given to one node only and the diameter of the graph is $\Theta(n)$.

such that the probability distribution of the tokens sent by them does not depend on the dynamic graph. Thus, if the algorithm terminates by round $r^* = \left\lfloor \frac{(n-2\lceil 3(R+\epsilon)\rceil+2)k}{2L\lfloor R \rfloor} \right\rfloor - 1$, then by linearity of expectation and Markov's inequality, there must exist some infrequently sent token t which is sent in $< \frac{n-2\lceil 3(R+\epsilon)\rceil+2}{L}$ rounds by all nodes $v_0, v_{n-1}, \dots, v_{n-\lfloor R \rfloor+3}$ with probability $> \frac{1}{2}$.

When one of the nodes $v_0, v_{n-1}, \dots, v_{n-\lfloor R \rfloor+3}$ sends token t for the first time, the adversary starts moving down lines 0, 1, and 2 until they reach levels 1, 2, and 3, respectively. These levels can be reached in $\left\lceil \frac{R+\epsilon}{2v_{\max}} \right\rceil$ rounds and by choice of L, token t cannot reach node $v_{n-2\lceil 3(R+\epsilon)\rceil+3}$. This guarantees that the nodes $v_0, v_{n-1}, \dots, v_{n-\lfloor R \rfloor+3}$ are again the only nodes knowing token t and being connected to the nodes at levels 0, 1, and 2. When one of the nodes $v_0, v_{n-1}, \dots, v_{n-\lfloor R \rfloor+3}$ sends token t for the second time, lines 3, 2, 1, and 0 are moved to levels 1, 2, 3, and 4, respectively. Again, by choice of L, token t cannot reach node $v_{n-2\lceil 3(R+\epsilon)\rceil+3}$. In general, when token t is sent for the $(j-1)^{\text{th}}$ time, then lines $j+1, j, j-1, j-2$, and $j-3$ are moved to levels 1, 2, 3, 4, and 5, respectively. The situation before token t is sent for the $j-1^{\text{th}}$ time is depicted in Figure 3.3.

As before, the connectivity graph is connected in each round. Because the nodes $v_0, v_{n-1}, \dots, v_{n-\lfloor R \rfloor+3}$ send token t in less than $\frac{n-2\lceil 3(R+\epsilon)\rceil+2}{L}$ rounds, it cannot reach node $v_{n-2\lceil 3(R+\epsilon)\rceil+3}$ and thus the algorithm cannot be finished by round $r^* = \Omega\big(nk \cdot \min\{v_{\max}, R\} \cdot R^{-3}\big)$ which concludes the proof of the theorem. □

Thus, it is possible to conclude that geometric dynamic networks with constant communication range and speed are as hard for knowledge-based token-forwarding algorithms as general dynamic networks.

Corollary 3.7 (Geometric Dynamic Networks with Constant Communication Ranges and Speeds)**.** *In geometric dynamic networks with communication range $R = \mathrm{O}(1)$ and $v_{\max} = \Omega(1)$, which are controlled by a weakly adaptive adversary, any knowledge-based token-forwarding algorithm requires $\Omega(nk)$ rounds to solve the k-token dissemination problem with probability $> \frac{1}{2}$.*

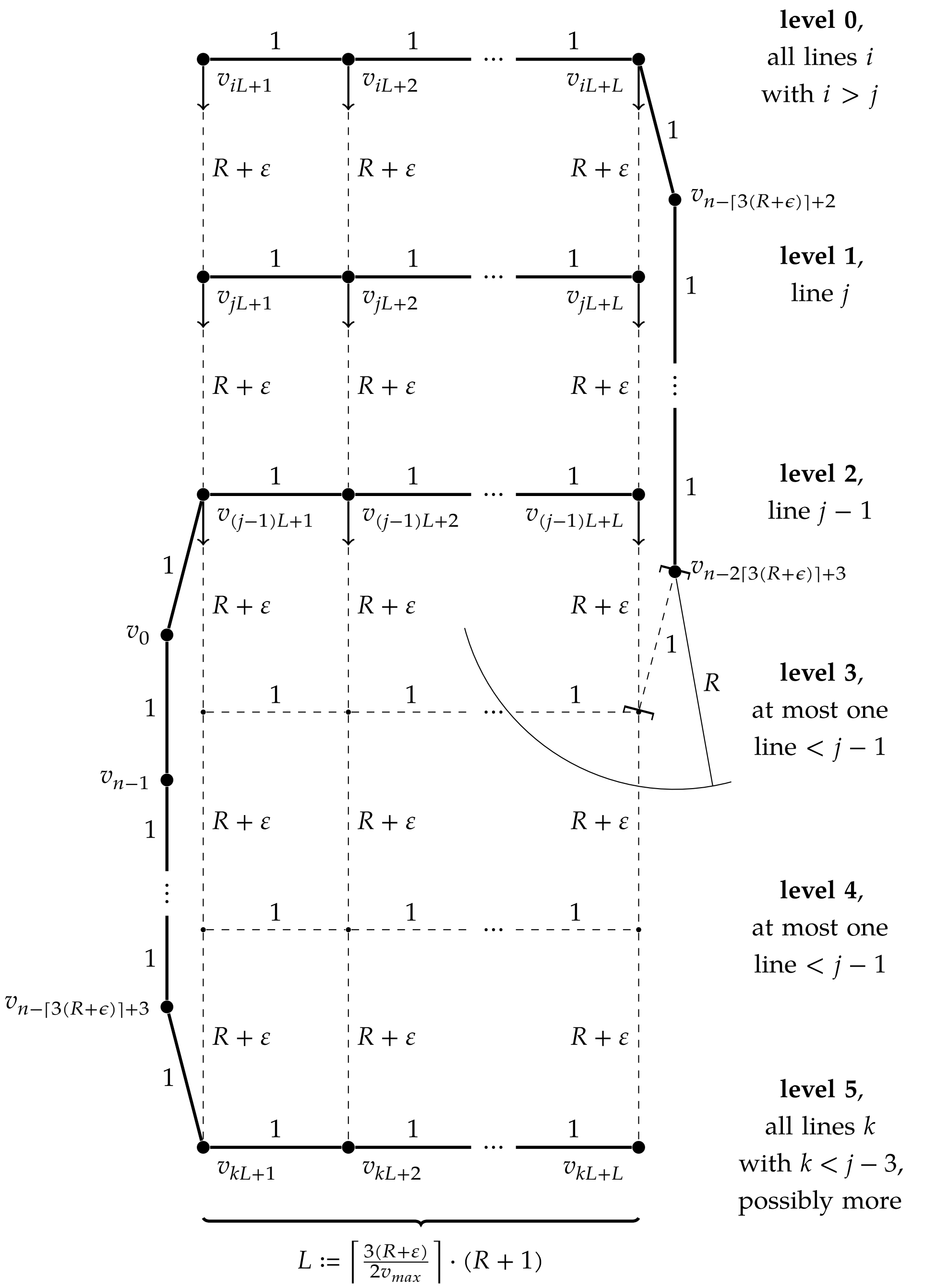

Figure 3.3: Lower bound construction for $R = 1$. Positions of all nodes after the infrequently sent token was sent for the $j-1^{\text{th}}$ time.

3.5 Conclusion and Unanswered Questions

In this chapter, a very natural dynamic network model has been introduced. As it turns out, both model parameters, the transmission range R and the maximum speed $v_{\max}$, influence the performance of knowledge-based token-forwarding algorithms solving the k-token dissemination problem. For the upper bounds, stable and vertex connected subgraphs can be used to achieve a speedup of the k-token dissemination algorithm. For the lower bounds, it is possible to define a dynamic UDG such that knowledge-based token-forwarding algorithms require many rounds to finish.

The results leave some questions unanswered. First of all, there is a gap of $\Theta(R)$ between the lower and the upper bound. It would be interesting to see whether the existing algorithm can be further improved or whether the lower bound can be strengthened to match the upper bound. Looking at the lower bound, it seems that a factor of $\Theta(R)$ is lost because it is assumed that the $\lfloor R \rfloor$ nodes in the cut could send the infrequently sent token in different rounds. Given that these nodes could send a different token based on their own unique ID only, it is not clear whether it is possible either to find an ID assignment such that they sent this token almost in the same round, or, if the unique ID can be used in the dissemination algorithm, to send a different token in many rounds.

Then, there are two unanswered questions related to the upper bound for token-forwarding algorithms. On the one hand, it requires that $R > 1$. Even for a centralized token-forwarding algorithm, it is an intriguing question whether a speed trade-off can be observed when $R = 1$. On the other hand, it is assumed that both R and $v_{\max}$ are known before the execution of the algorithm. Both parameters affect the interval stability which can be learned (as shown by Kuhn et al. [KLO10]), but R also influences the C-vertex connectivity. While it is possible to upper bound C depending on the number of neighbors, different nodes could observe different numbers.

Finally, it would also be interesting to move the attention from token-forwarding algorithms to more advanced techniques such as network coding based approaches and find out how these perform in the geometric dynamic network model.

CHAPTER 4

Counting versus Token Dissemination

One of the most fundamental problems, one could ask to solve in a dynamic network, is the problem of counting all its nodes. As counting reduces to the problem of disseminating all unique IDs (and then counting them locally), any algorithm that solves the all-to-all token dissemination problem can be used to solve the counting problem. The best known counting algorithm is the algorithm by Kuhn et al. [KLO10] (cf. Section 2.4.1). It implicitly disseminates all n unique IDs to all nodes and requires $\Theta(n^2)$ rounds in 1-stable / 1-interval connected dynamic networks. Existing lower bounds concerning the token dissemination problem (such as those for token-forwarding algorithms) suggest that algorithms which solve both counting and all-to-all token dissemination cannot be essentially faster. Nevertheless, the question, whether it is possible to solve the counting problem faster, remains unanswered.

Unanswered Question 4.1 (Complexity of Counting)**.** Is it possible to solve the counting problem in $o(n^2)$ rounds in 1-stable / 1-interval connected dynamic networks?

This chapter establishes another relation between the counting problem and token dissemination problems in directed dynamic networks. For this, a specialization of the k-token dissemination problem, the two-party k-token dissemination problem, is introduced, in which one node initially receives k tokens that have to be disseminated to one predefined communication partner only. Through a series of model reductions, it is shown that solving a two-party $\Theta(n)$-token dissemination problem in a directed dynamic network with $\Theta(n)$ nodes requires at most as many rounds as counting the number of nodes in a directed dynamic network with $\Theta(n)$ nodes. Therefore, in order to find a lower bound for the counting problem in directed dynamic networks, it is sufficient to find a lower bound for two-party $\Theta(n)$-token dissemination problems. Or interpreted differently, if there is a fast counting algorithm, then there must exist a fast algorithm for the two-party $\Theta(n)$-token dissemination problem.

Chapter Basis The results presented in this chapter are based on the following publication.

> **2013** (with M. Benter, M. Malatyali, and F. Meyer auf der Heide).
> "On Two-Party Communication through Dynamic Networks."
> In: *Principles of Distributed Systems - 17th International Conference, OPODIS 2013, Nice, France, December 16-18, 2013. Proceedings*, cf. [Abs+13b].

The proofs in this chapter are revised and state the dependencies between the cardinalities of the unique ID and token universes and the message sizes more precisely.

Chapter Outline Section 4.1 introduces a two-party variant of the k-token dissemination problem and clarifies the assumptions about the class of algorithms studied in this chapter. Related work with respect to the counting problem and to the techniques of this chapter is reviewed in Section 4.2. Section 4.3 establishes a relation between the counting problem and the so-called same predecessors problem in a special class of directed dynamic networks. In Section 4.4, the same predecessors problem is related to the set equality problem in so-called dynamic channel networks, which is another special class of directed dynamic networks. Next, Section 4.5 gives lower bounds for the set equality problem in the two-party communication model by Yao. These findings are then used in Section 4.6 to relate the set equality problem to the two-party token dissemination in directed dynamic networks. Finally, Section 4.7 combines the intermediate results of the preceding sections and states the main result. Section 4.8 concludes this chapter.

4.1 Models & Problems

This chapter addresses the directed version of dynamic networks as defined in Section 2.2. The following problem is a specialization of the k-token dissemination problem which fixes the initial token assignment to a single node and requires the dissemination of the tokens to one node only.

Problem 4.2 (Two-Party k-Token Dissemination)**.** One node v_s is given a subset of cardinality k from some known token universe $\mathcal{T} \cong [\mathrm{poly}(n)]$ as input $I(v_s)$. All other nodes do not receive any input. The nodes of the network should disseminate these k tokens such that a given node v_t decides on $I(v_s)$. The parameter k may be known to the nodes beforehand.

If n is known, then this problem can easily be solved in $\mathrm{O}(nk)$ rounds in 1-stable / 1-interval connected dynamic networks (cf. Section 2.4.1).

4.1.1 Token and Unique ID Universes and Message Sizes

Recall that the cardinalities of the token and ID universes are assumed to be polynomial in n and that the message size is limited by a number of bits that is logarithmic in n and large enough to encode one unique ID or one token. To apply communication theoretic arguments, these assumptions have to be stated more precisely.

It is assumed that the algorithms examined in this chapter (and most parts of this thesis) are uniform in the sense that they do not know the number of nodes in the beginning. However, to be able to send a message, a node must know the maximum message size which can be viewed as an additional input at the beginning. Furthermore, it is assumed that a node knows the cardinalities of the unique ID and token universes.[1] All these additional inputs are just numbers that draw almost no conclusions about the number of nodes, i.e., a

[1] This is reasonable because unique identifiers are usually represented as bit strings, e.g., according to IEEE Std 802-2014 [14], MAC addresses are 48-bit strings and thus a node may conclude that there are less than 2^{48} nodes in total.

node is not able to calculate the exact value of n based on these numbers only, since the exact polynomials and logarithms are not known. The algorithms considered in this chapter are expected to work for all polynomial unique ID and token universes and all logarithmic message sizes.

4.2 Related Work

The counting problem and the naming problem have been studied by Michail et al. [MCS12a; MCS13] in anonymous dynamic networks. Here, the naming problem is to find unique IDs in a variant of dynamic networks, in which nodes do not have unique IDs in the beginning. Even in anonymous static networks, it is impossible to count without a leader. Naming with a leader and given the number of nodes is impossible too. However, given a leader, counting in anonymous static networks becomes possible. In anonymous dynamic networks, the authors conjecture that it is impossible to compute anything nontrivial, even with a leader. Therefore, also counting is conjectured to be impossible. Furthermore, they show that under additional assumptions, such as nodes knowing an upper bound on the maximum degree, it becomes possible to compute an upper bound on the number of nodes. Since some of the impossibility results stem from the broadcast communication model, they introduced another model called *one-to-each message transmission*, in which nodes are bestowed with the ability to send different messages to different neighbors. In the presence of the leader, naming (and thus counting) becomes possible with one-to-each message transmission.

Using the technique by Michail et al. [MCS13] to obtain an upper bound on the number of nodes if a leader is present and an upper bound on the maximum degree is known, Di Luna et al. [Di +14a; Di +13] introduce an algorithm to obtain the exact count in anonymous dynamic networks. Their techniques are inspired by the idea of an energy-transfer between the nodes. In a different paper, Di Luna et al. [Di +14b] show how to count in anonymous dynamic networks, in which a leader is present and the nodes are equipped with different oracles that give information about the node degrees.

In static networks, Das Sarma et al. [Das+12] apply techniques similar to those used in this chapter. They apply lower bounds from communication theory to obtain lower bounds for distributed algorithms for many fundamental graph problems such as the verification of minimum spanning trees. Subsequently, Frischknecht et al. [FHW12] show that static networks require $\Omega(n/\log n)$ rounds to compute their diameter. In directed static networks, Kuhn and Oshman [KO11b] show that problems, such as counting, require $\Omega(n/\log n)$ rounds even if the diameter of the network is 2.

4.3 Same Predecessors & Counting

First, a relation between a problem defined on a special class of directed dynamic networks and counting in directed dynamic networks is established.

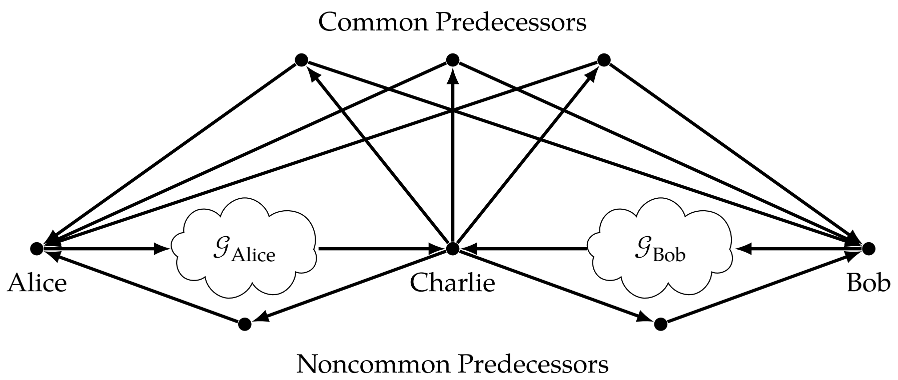

Figure 4.1: Construction of a Special Dynamic Network $\mathcal{G}_d$

A *special dynamic network parameterized by d* is a directed dynamic network $\mathcal{G}_d$ that consists of n nodes with $3d + 3 \leq n \leq 4d + 3$ and is defined as follows: $\mathcal{G}_d$ consists of three special nodes Alice, Bob, and Charlie and any two 1-stable / 1-interval connected directed dynamic networks $\mathcal{G}_{\text{Alice}}$ and $\mathcal{G}_{\text{Bob}}$, each with exactly d nodes. Alice and Bob have one outgoing edge to one fixed node $v_{s,\text{Alice}}$ in $\mathcal{G}_{\text{Alice}}$ and $v_{s,\text{Bob}}$ in $\mathcal{G}_{\text{Bob}}$, respectively. $\mathcal{G}_{\text{Alice}}$ and $\mathcal{G}_{\text{Bob}}$ have one outgoing edge to Charlie from one fixed node $v_{t,\text{Alice}}$ and $v_{t,\text{Bob}}$, respectively. In addition, $\mathcal{G}_d$ consists of further d to $2d$ nodes that are predecessors of Alice or Bob such that the in-degrees of Alice and Bob are exactly d. This means that if there are exactly d predecessors, then these d nodes are predecessors of both Alice and Bob, and if there are more than d predecessors, then at least one node is a predecessor of either Alice or Bob. Nodes that are predecessors of both Alice and Bob are called *common predecessors*. Nodes that are predecessors of either Alice or Bob are called *noncommon predecessors*. Charlie is connected to all common and noncommon predecessors. The edges in $\mathcal{G}_{\text{Alice}}$ and $\mathcal{G}_{\text{Bob}}$ are dynamic, all other edges remain static. The construction of $\mathcal{G}_d$ is depicted in Figure 4.1.

$\mathcal{SDN}_d$ is defined as the set of all special dynamic networks parameterized by d. The set of all special dynamic networks is defined as $\mathcal{SDN} := \bigcup_{d \in \mathbb{N}^+} \mathcal{SDN}_d$. All nodes know that the dynamic network is a special dynamic network, and Alice and Bob know their role, i.e., they know that they are Alice and Bob, respectively. All other nodes do not know their role, in particular, the predecessors do not know whether they are common or noncommon predecessors. The total number of nodes in the special dynamic network n is not known beforehand. Furthermore, the parameter d is also not known beforehand but can be easily obtained by Alice and Bob (cf. Lemma 4.4). The following problem is defined on special dynamic networks in $\mathcal{SDN}$.

Definition 4.3 (Same Predecessors Problem SP)**.** The same predecessors function SP : $\mathcal{SDN} \to \{0, 1\}$ is defined as

$$\text{SP}(\mathcal{G}_d) = \begin{cases} 1 & \text{if } N^-(\text{Alice}) = N^-(\text{Bob}) \\ 0 & \text{otherwise} \end{cases} .$$

Given a counting algorithm for general directed dynamic networks, the following lemma shows how to solve the same predecessors problem in special dynamic networks.

Lemma 4.4 (Same Predecessors versus Counting)**.** *Let T^*_{SP} be the minimum number of rounds required by all deterministic algorithms that decide* SP *in special dynamic networks of size at most n. Let T_{C} be the number of rounds any deterministic distributed algorithm requires to decide the counting problem in a general directed dynamic network of size at least n with the same message size and unique ID universe. Then,*

$$T^*_{\text{SP}} \leq T_{\text{C}} + n.$$

Proof. Alice and Bob are able to determine their in-degree d if each node sends a message in the first round. Therefore, Alice and Bob know that the special dynamic network is in $\mathcal{SDN}_d$. Given the counting algorithm, all nodes can determine the total number of nodes n in the network in further T_{C} rounds.

Now, by construction of the special dynamic network, Alice and Bob have the same predecessors if and only if $n = 3 + 3d$ and thus, Alice and Bob can decide SP. The result available at Alice and Bob can be disseminated in the whole network in at most $n - 1$ rounds. □

4.4 Set Equality & Same Predecessors

Next, the same predecessors problem is related to the problem of deciding whether Alice and Bob are given the same subsets of elements from some set $\mathcal{S}$ in another directed dynamic network.

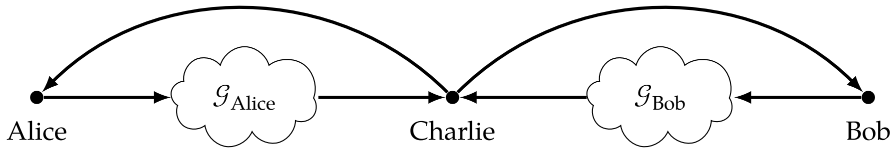

Figure 4.2: Construction of a Dynamic Channel Network $\mathcal{G}'_d$

A *dynamic channel network parameterized by d* is a dynamic network $\mathcal{G}'_d$ defined in a similar way compared to the special dynamic networks. It consists of exactly $n = 2d + 3$ nodes. As before, there are three special nodes Alice, Bob, and Charlie and any two 1-stable / 1-interval connected dynamic networks $\mathcal{G}_{\text{Alice}}$ and $\mathcal{G}_{\text{Bob}}$, each with exactly d nodes. Again, Alice and Bob have one outgoing edge to one fixed node $v_{s,\text{Alice}}$ in $\mathcal{G}_{\text{Alice}}$ and $v_{s,\text{Bob}}$ in $\mathcal{G}_{\text{Bob}}$, respectively, and $\mathcal{G}_{\text{Alice}}$ and $\mathcal{G}_{\text{Bob}}$ have one outgoing edge to Charlie from one fixed node $v_{t,\text{Alice}}$ and $v_{t,\text{Bob}}$, respectively. Now, Charlie is directly connected to Alice and Bob. Similarly, the edges in $\mathcal{G}_{\text{Alice}}$ and $\mathcal{G}_{\text{Bob}}$ are dynamic, all other edges remain static. The construction of $\mathcal{G}'_d$ is depicted in Figure 4.2.

$\mathcal{DCN}_d$ is defined as the set of all dynamic channel networks parameterized by d. The set of all dynamic channel networks is defined as $\mathcal{DCN} := \bigcup_{d \in \mathbb{N}^+} \mathcal{DCN}_d$. All nodes know that the dynamic network is a dynamic channel network, and all nodes know their role, i.e., they know whether they are Alice, Bob, Charlie, or a node in $\mathcal{G}_{\text{Alice}}$ or $\mathcal{G}_{\text{Bob}}$. Here, the nodes may know n.

The following set equality problem is defined between two communication partners that are given two sets A and B of elements from some set $\mathcal{S}$ of polynomial size. These communication partners have to decide whether they are given the same sets of elements.

Definition 4.5 (Set Equality Problem EQ_n)**.** Let T_n be the set of all subsets of cardinality n of a set $\mathcal{S} \cong [\mathrm{poly}(n)]$. For $A, B \in T_n$, the set equality function $\mathrm{EQ}_n : T_n \times T_n \to \{0,1\}$ is defined as

$$\mathrm{EQ}_n(A,B) = \begin{cases} 1 & \text{if } A = B \\ 0 & \text{otherwise} \end{cases}.$$

Given an algorithm for the same predecessors problem, Alice and Bob can solve the set equality problem in dynamic channel networks as the following lemma shows.

Lemma 4.6 (Set Equality versus Same Predecessors)**.** *Let $T^*_{\mathrm{EQ}_n}$ be the minimum number of rounds required by all deterministic algorithms that decide* EQ_n *in dynamic channel networks of size at most n and unique ID universe $\mathcal{U}$. Let T_{SP} be the number of rounds any deterministic distributed algorithm requires to decide* SP *in a special dynamic network of size at least $2n-3$, the same message size, and a unique ID universe $\mathcal{U}'$ large enough such that $|\mathcal{U}'| \geq |\mathcal{U}| + |\mathcal{S}|$. Then,*

$$T^*_{\mathrm{EQ}_n} \leq T_{\mathrm{SP}}.$$

Proof. Assume there is an algorithm that solves the same predecessors problem SP in T_{SP} rounds. Since the computational power of each node is assumed to be unbounded, Alice and Bob in the dynamic channel network can simulate the algorithm executed on Alice and Bob and on a virtual predecessor node for each element of A and B, respectively. Messages Alice and Bob receive from Charlie are passed on to the simulated nodes, messages sent by the simulated nodes are passed on to Alice and Bob simulated in the special dynamic network and messages sent by Alice and Bob are sent to the dynamic networks $\mathcal{G}_{\mathrm{Alice}}$ or $\mathcal{G}_{\mathrm{Bob}}$. All other nodes can directly execute the algorithm. For the unique ID universe, the disjoint union of the unique ID universe in the dynamic channel network and $\mathcal{S}$ from the EQ_n problem is used, i.e., during simulation, the unique ID used for each node u in the dynamic channel network is $(\mathrm{id}(u), 0)$ and the unique ID used for each node simulated by Alice (and Bob) for each

$t \in A$ (and $t \in B$) is $(t, 1)$. Note that the cardinality of this unique ID universe is still polynomial in n and therefore the algorithm for the same predecessors problem requires a message size that is at most a logarithmic in n. Since Alice and Bob have the same predecessors in the special dynamic network if and only if Alice and Bob have the same sets in the dynamic channel network, they can decide the EQ_n problem in at most T_{SP} rounds. □

4.5 Two-Party Communication & Set Equality

The last model is used to bound the number of bits that have to be exchanged between Alice and Bob to solve the set equality problem EQ_n.

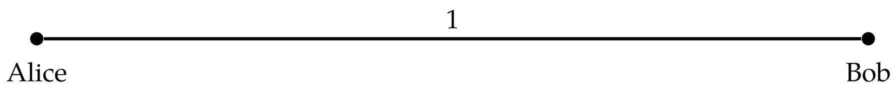

Figure 4.3: Two-Party Communication

In the two-party communication model by Yao [Yao79], two communication partners Alice and Bob are given inputs A and B, respectively. They are connected by an undirected edge which they can use to alternately exchange 1-bit messages in order to compute a function $f(A, B)$. In this model, two rounds correspond to at least one round if Alice and Bob are allowed to send messages at the same time. The model is depicted in Figure 4.3.

Yao [Yao79] introduced a decomposition-based technique to obtain a lower bound on the number of rounds necessary to compute those functions: For a function $f : M \times N \to \{0, 1\}$, the Cartesian product $S \times T$ (where $S \subseteq M$ and $T \subseteq N$) is called an *f-monochromatic rectangle* if f is constant over $S \times T$. A *k-decomposition of f* is a family $\{S_1 \times T_1, S_2 \times T_2, \dots, S_k \times T_k\}$ of k disjoint f-monochromatic rectangles that partition $M \times N$. The following theorem bounds the *two-way complexity* $C(f)$, i.e., the number of rounds required to solve f in Yao's two-party communication model. Its proof is based on bounding the minimum height of a decision tree for f.

Theorem 4.7 (Two-Way Complexity Lower Bound). *Let $f : M \times N \to \{0, 1\}$ be a function and let $d(f)$ be the minimum k such that a k-decomposition of f exists. Then, $C(f) \geq \log_2 d(f) - 2$. [Yao79]*

The next lemma applies the preceding theorem to the set equality problem.

Lemma 4.8 (Two-Way Complexity Set Equality Problem)**.** *In the two-party communication model by Yao, any deterministic algorithm for* EQ_n *requires at least*

$$C'(\mathrm{EQ}_n) := \log_2 \binom{|\mathcal{S}|}{n} - 4$$

rounds for at least $\frac{1}{2}\binom{|\mathcal{S}|}{n}$ *inputs.*

Proof. EQ_n is a Boolean function on $T_n \times T_n$. Consider a monochromatic rectangle $\tilde{A} \times \tilde{B}$ of EQ_n evaluating to 1. Note that $|\tilde{A} \times \tilde{B}| = 1$ since otherwise there would be $x \in \tilde{A}$ and $y \in \tilde{B}$ with $x \neq y$, and thus, $\mathrm{EQ}_n(x, y) = 0$. Hence, the number of EQ_n-monochromatic rectangles with result 1 is exactly the number of inputs (X, X) with $X \in T_n$, and therefore,

$$\begin{aligned} & \#\mathrm{EQ}_n\text{-monochromatic rectangles} \\ \geq\ & \#\mathrm{EQ}_n\text{-monochromatic rectangles with result 1} \\ =\ & |T_n| = \binom{|\mathcal{S}|}{n}. \end{aligned}$$

By Theorem 4.7,

$$C(\mathrm{EQ}_n) \geq \log_2 \binom{|\mathcal{S}|}{n} - 2.$$

A similar argument shows that for at least $\frac{1}{2}\binom{|\mathcal{S}|}{n}$ inputs at least

$$C'(\mathrm{EQ}_n) = \log_2 \binom{|\mathcal{S}|}{n} - 4$$

rounds are required: If the decision tree is cut at level $C'(\mathrm{EQ}_n)$, it contains less than $\frac{1}{2}\binom{|\mathcal{S}|}{n}$ leaves. Hence, all other leaves must have distance greater than $C'(\mathrm{EQ}_n)$ to the root of the tree. □

The previous lemma answers how many rounds are needed to compute EQ_n in Yao's two-party communication model. To transfer these insights to the dynamic channel model, the following lemma states how many of these 1-bit messages are essential in the sense that either Alice has to send them to Bob or Bob has to send them to Alice. It also states that the possibility of sending larger messages—such as messages of logarithmic size in the dynamic channel model—does not reduce the amount of bits that have to be exchanged.

Lemma 4.9 (One-Way Communication Set Equality Problem)**.** *In order to decide* EQ_n*, either Alice has to send at least*

$$\frac{1}{4}C'(\mathrm{EQ}_n) = \frac{1}{4}\log_2\binom{|\mathcal{S}|}{n} - 1$$

bits to Bob or vice versa for at least $\frac{1}{2}\binom{|\mathcal{S}|}{n}$ *inputs. This also holds if Alice and Bob are allowed to send larger messages through the communication channel between them.*

Proof. Assume to the contrary that in order to solve EQ_n both Alice and Bob have to send $< \frac{1}{4}C'(\mathrm{EQ}_n)$ 1-bit messages to Bob and Alice, respectively. Then, in total $< \frac{1}{2}C'(\mathrm{EQ}_n)$ 1-bit messages are needed. Consequently, $< C'(\mathrm{EQ}_n)$ rounds are required if 1-bit messages have to be sent alternately as in Yao's model which contradicts Lemma 4.8.

Now, if Alice and Bob are allowed to send larger messages and thereby group multiple 1-bit messages, then this could only increase the number of bits needed to solve the problem since larger messages only reduce the ability to interact for Alice and Bob. □

4.6 Two-Party Token Dissemination & Set Equality

Knowing that Alice and Bob have to exchange many bits, it is possible to relate the two-party token dissemination problem to the set equality problem.

Lemma 4.10 (Two-Party Token Dissemination versus Set Equality)**.** *Let* T^*_{D} *be the minimum number of rounds required by all deterministic algorithms that solve the two-party n-token dissemination problem in directed dynamic networks of size at most n. Let* T_{EQ_n} *be the number of rounds any deterministic distributed algorithm requires to decide* EQ_n *in dynamic channel networks of size at least* $2n + 3$*, and with the same message size, the same unique ID universe and* $\mathcal{S}$ *large enough such that*

$$\log_2\binom{|\mathcal{S}|}{n} \geq 4\log_2\binom{|\mathcal{T}|}{n} + 4.$$

Then,

$$T^*_{\mathrm{D}} \leq T_{\mathrm{EQ}_n}.$$

Proof. If T_{D}^{*} rounds are required to solve the two-party n-token dissemination problem in directed dynamic networks of size n such as $\mathcal{G}_{\mathrm{Alice}}$ and $\mathcal{G}_{\mathrm{Bob}}$, then it is particularly not possible to send any $\log_2 \binom{|\mathcal{T}|}{n}$-bit string from $v_{s,\mathrm{Alice}}$ to $v_{t,\mathrm{Alice}}$ (and $v_{s,\mathrm{Bob}}$ to $v_{t,\mathrm{Bob}}$) in less than T_{D}^{*} rounds since this string could be used to encode the n tokens given to $v_{s,\mathrm{Alice}}$ (or $v_{t,\mathrm{Bob}}$). However, by Lemma 4.9, $\frac{1}{4}\log_2 \binom{|\mathcal{S}|}{n} - 1$ bits have to be sent from Alice to Bob or vice versa, in order to solve EQ_n. Hence, more bits have to be sent through the dynamic network for solving EQ_n than for solving the two-party n-token dissemination problem. Thus, $T_{\mathrm{D}}^{*} \leq T_{\mathrm{EQ}_n}$. □

4.7 Two-Party Token Dissemination & Counting

Now, all the lemmas from the preceding sections can be combined to relate the two-party token dissemination problem to the counting problem.

Theorem 4.11 (Two-Party Token Dissemination versus Counting)**.** *Let T_{D}^{*} be the minimum number of rounds required by all deterministic algorithms that solve the two-party n-token dissemination problem in directed dynamic networks of size at most n, unique ID universe $\mathcal{U}$ and token universe $\mathcal{T}$. Let T_{C} be the number of rounds any deterministic distributed algorithm requires to decide the counting problem in a general directed dynamic network of size at least $4n + 3$ with the same message size and a unique ID universe $\mathcal{U}'$ large enough such that*

$$\log_2 \binom{|\mathcal{U}'| - |\mathcal{U}|}{n} \geq 4\log_2 \binom{|\mathcal{T}|}{n} + 4.$$

Then,

$$T_{\mathrm{D}}^{*} \leq T_{\mathrm{C}} + 4n + 3.$$

Proof. Let $T_{\mathrm{EQ}_n}^{*}$ be the minimum number of rounds required by all deterministic algorithms that decide EQ_n in dynamic channel networks of size at most $2n + 3$ and with the same message size and the same unique ID universe as for the two-party token dissemination problem and $\mathcal{S}$ large enough such that $\log_2 \binom{|\mathcal{S}|}{n} \geq 4\log_2 \binom{|\mathcal{T}|}{n} + 4$. Then, by Lemma 4.10, $T_{\mathrm{D}}^{*} \leq T_{\mathrm{EQ}_n}^{*}$.

Let T_{SP}^{*} be the minimum number of rounds required by all deterministic algorithms that decide SP in special dynamic networks of size at most $4n + 3$, the same message size and a unique ID universe that is at least as large as the unique ID universe in the dynamic channel network and $\mathcal{S}$. Then, by Lemma 4.6, $T^*_{\mathrm{EQ}_n} \leq T^*_{\mathrm{SP}}$.

Finally, let T_{C} be the number of rounds any deterministic distributed algorithm requires to decide the counting problem in a general directed dynamic network of size $4n + 3$ with the same message size and unique ID universe. Then, by Lemma 4.4, $T^*_{\mathrm{SP}} \leq T_{\mathrm{C}} + 4n + 3$. □

Using this theorem, it is possible to transfer a super-linear lower bound for the two-party $\Theta(n)$-token dissemination problem to the counting problem.

Corollary 4.12 (Lower Bound for Counting)**.** *If the two-party $\Theta(n)$-token dissemination problem in a directed dynamic network of size $\Theta(n)$ requires a super-linear number of rounds, then counting in directed dynamic networks of size $\Theta(n)$ requires a super-linear number of rounds as well. The unique ID universe of the counting problem is only polynomially larger than the token universe and the ID universe of the dissemination problem.*

Stated differently, given that there is a fast, i.e., sub-quadratic, counting algorithm, this means that there must exist a fast algorithm for the two-party $\Theta(n)$-token dissemination problem.

Corollary 4.13 (Upper Bound for Two-Party Token Dissemination)**.** *If the counting problem in a directed dynamic network size $\Theta(n)$ can be solved in a sub-quadratic number of rounds, then the two-party $\Theta(n)$-token dissemination problem in a directed dynamic network of size $\Theta(n)$ can be solved in a sub-quadratic number of rounds as well. The unique ID universe of the counting problem is only polynomially larger than the token universe and the ID universe of the dissemination problem.*

This directly yields the following unanswered question.

Unanswered Question 4.14 (Complexity of Two-Party Token Dissemination)**.** Is it possible to solve the two-party n-token dissemination problem in $o(n^2)$ rounds in 1-stable / 1-interval connected dynamic networks?

4.8 Conclusion and Unanswered Questions

This chapter established a relation between the counting problem and a two-party variant of the n-token dissemination problem. While the question about the exact complexity of counting remains unanswered, and instead another question about the complexity of the two-party n-token dissemination problem arises, the hope is that the latter is easier to answer since existing results focus rather on token dissemination problems and on counting.

The question about the complexity of the two-party n-token dissemination problem is challenging: On the one hand, finding a nontrivial upper bound seems to be difficult since it is not clear how a dissemination algorithm could exploit the knowledge that v_t is the only node that has to receive all tokens. On the other hand, existing lower bounding techniques for n-token dissemination cannot be directly applied to this special subproblem.

Since the construction presented in this chapter requires directed edges, it would be interesting to see if a similar relation can be shown in undirected dynamic networks or if the complexity of the counting problem is different in undirected networks.

CHAPTER 5

Continuous Aggregation in Dynamic Networks

SENSOR NETWORKS observe the environment to which they are deployed and build up ad-hoc networks to transmit their data. Energy is a scarce resource in these networks. Thus, in order to conserve battery power, data is usually transmitted to some sink node, either to process and store it there or to upload it to a remote location from there involving high communication costs. For this, data is often aggregated along trees rooted in the sink node. Typical examples of aggregation functions are the minimum or the maximum, the average, the medium or the mode, or the sum. These techniques work well and reduce the amount of communication if the network is stable.

This chapter addresses a more dynamic scenario and studies sensor networks from the perspective of T-stable dynamic networks. Compared to the previously described approaches for static networks, the minimization of the communication amount needed by the distributed algorithms is not the primary optimization goal but rather solving problems efficiently, i.e., fast, under given and fixed communication restrictions. As dynamic networks are constantly changing, it seems to be hard to route aggregated information to a single sink node only (cf. Chapter 4). Therefore, in this chapter, all nodes—particularly sink nodes—should be aware of the result. Nevertheless, the idea of aggregation can be transferred and beyond that pipelining techniques give a speedup for continuous computation. Here, *continuous computation* refers to the requirement of an aggregation function not being applied to the measured data of some point in time only once, but again and again for multiple points in time.

In this chapter, algorithms for the continuous computation of the extremum (e.g., the maximum) and the summation (e.g., the sum) are presented. To evaluate their performance, two performance metrics are introduced. The first metric is the *delay*, which is defined as the maximum number of rounds between the round when inputs arrive at all nodes and the round the last node outputs the result of the aggregation function of these inputs. The second metric is the *output rate*, which measures the number of outputs generated in a time interval divided by the length of this interval (see Section 5.1 for a formal definition of this metric).

In static networks, the noncontinuous extremum problem, the noncontinuous summation problem, and the (noncontinuous) all-to-all token dissemination problem can be solved in a linear number of rounds (cf. Section 5.3). The continuous variants of the extremum and summation problems can be solved with a constant output rate. The continuous all-to-all token dissemination problem can be solved with output rate $\Omega\left(\frac{1}{n}\right)$, while all delays remain linear. Note that these results are asymptotically tight in static networks.

For 1-stable dynamic networks, it is shown that the noncontinuous extremum problem can still be solved in a linear number of rounds (cf. Section 5.4.2). To solve the other problems, it is assumed that $T \geq c \cdot \mathrm{MIS}(n)$, where c is a sufficiently large constant and $\mathrm{MIS}(n)$ is the number of rounds required to compute a maximal independent set in a graph with n nodes.[1] Compared to the (noncontinuous) all-to-all token dissemination problem, it is possible to solve the noncontinuous summation problem $\Theta\left(\frac{T}{\mathrm{MIS}(n)}\right)$ times faster if $T = \mathrm{O}\left(\sqrt{n \cdot \mathrm{MIS}(n)}\right)$, and if $T = \Omega\left(\sqrt{n \cdot \mathrm{MIS}(n)}\right)$, this problem can be solved in a linear number of rounds.

For the continuous extremum and the summation problem, nontrivial output rates are proven, i.e., output rates that are higher than those obtained by executing the noncontinuous algorithms over and over again. For the con-

[1]Note that there are some restrictions under which this maximal independent set must be computed (cf. Section 5.4.1).

tinuous extremum problem, this increases the delay only slightly. If $T = O(n^{2/3}\,\mathrm{MIS}(n)^{2/3})$, it is possible to achieve the same delay and a slightly smaller output rate for the continuous summation problem compared to the extremum problem. Besides these deterministic results (cf. Table 5.1), it is shown in the corresponding sections how randomization helps to improve these results.

Chapter Basis The results presented in this chapter are based on the following publication.

> **2014** (with F. Meyer auf der Heide). "Continuous Aggregation in Dynamic Ad-Hoc Networks." In: *Structural Information and Communication Complexity - 21st International Colloquium, SIROCCO 2014, Takayama, Japan, July 23-25, 2014. Proceedings*, cf. [AM14].

Chapter Outline Section 5.1 defines the model, introduces the noncontinuous and continuous aggregation problems studied in this chapter, and explains the performance metrics used to evaluate the algorithms in this chapter. Related work with respect to aggregation problems in networks is surveyed in Section 5.2 as well as algorithms for computing a maximal independent set, which form an important building block of this chapter. Section 5.3 shows how the noncontinuous and continuous aggregation problems can be solved in static networks and demonstrates how the metrics can be applied. Then, Section 5.4 deals with T-stable dynamic networks. First, in Section 5.4.1, the graph patching technique is introduced. Then, the noncontinuous extremum and the noncontinuous summation problems are solved in Section 5.4.2 and Section 5.4.3, respectively. Then, the continuous variants are solved in Section 5.4.4 and Section 5.4.5, respectively. Section 5.5 shows how these results can be applied to geometric dynamic networks. Section 5.6 concludes the chapter and states unanswered questions.

Table 5.1: Overview: Deterministic Results.

(a) Static Networks.

	Extremum	**Summation**	**Dissemination**
noncontinuous			
Running Time:	$O(n)$	$O(n)$	$O(n)$
continuous			
Delay:	$O(n)$	$O(n)$	$O(n)$
Output Rate:	$\Omega(1)$	$\Omega(1)$	$\Omega\left(\frac{1}{n}\right)$

(b) T-Stable Dynamic Networks with $T \geq c \cdot \mathrm{MIS}(n)$.

	Extremum	**Summation**	**Dissemination**
noncontinuous			
Running Time:			
- if $T = O\left(\sqrt{n \cdot \mathrm{MIS}(n)}\right)$	$O(n)$	$O\left(\frac{n^2 \cdot \mathrm{MIS}(n)}{T^2}\right)$	$O\left(\frac{n^2}{T}\right)$
- if $T = \Omega\left(\sqrt{n \cdot \mathrm{MIS}(n)}\right)$		$O(n)$	
continuous			
Delay:			
- if $T = O(n^{2/3}\,\mathrm{MIS}(n)^{2/3})$	$O(n \cdot \mathrm{MIS}(n))$	$O\left(\frac{n^2 \cdot \mathrm{MIS}(n)^2}{T^{3/2}}\right)$	$O\left(\frac{n^2}{T}\right)$
- if $T = \Omega(n^{2/3}\,\mathrm{MIS}(n)^{2/3})$		$O(n \cdot \mathrm{MIS}(n))$	
Output Rate:			
- if $T = O(n^{2/3}\,\mathrm{MIS}(n)^{2/3})$	$\Omega\left(\frac{T}{n \cdot \mathrm{MIS}(n)}\right)$	$\Omega\left(\frac{T^{5/2}}{n^2 \cdot \mathrm{MIS}(n)^3}\right)$	$\Omega\left(\frac{T}{n^2}\right)$
- if $T = \Omega(n^{2/3}\,\mathrm{MIS}(n)^{2/3})$		$\Omega\left(\frac{T}{n \cdot \mathrm{MIS}(n)^2}\right)$	

5.1 Models & Problems

In this chapter, the following two noncontinuous aggregation problems are studied in the undirected, T-stable dynamic network model where both n and T are known (cf. Section 2.2).

Problem 5.1 (Noncontinuous Extremum). Let $(S, \oplus)$ be a commutative and idempotent semigroup and let the elements of S be representable by $\mathrm{O}(\log n)$ bits. Each node i in the network receives an element $x_i \in S$ as input. Let f be defined by

$$f(x_1, x_2, \ldots, x_n) := \bigoplus_{i=1}^{n} x_i.$$

All nodes of the network have to output $f(x_1, x_2, \ldots, x_n)$.

For example, consider the minimum or maximum functions over a subset of $\mathbb{N}_{\geq 0}$ of size polynomial in n, i.e., $S = [\mathrm{poly}(n)]$ and $\oplus : S \times S \to S, (x, y) \mapsto \min\{x, y\}$ or $\oplus : S \times S \to S, (x, y) \mapsto \max\{x, y\}$. Other examples are the bitwise *and* and the bitwise *or* over bit strings of length logarithmic in n, i.e., $S = \{0, 1\}^{\mathrm{O}(\log n)}$ and $\oplus : S \times S \to S, (x, y) \mapsto x \,\&\, y$ or $\oplus : S \times S \to S, (x, y) \mapsto x \mid y$.

Problem 5.2 (Noncontinuous Summation). Let $(S, \oplus)$ be a commutative semigroup and let the elements of S be representable by $\mathrm{O}(\log n)$ bits. Each node i in the network receives an element $x_i \in S$ as input. Let f be defined by

$$f(x_1, x_2, \ldots, x_n) := \bigoplus_{i=1}^{n} x_i.$$

All nodes of the network have to output $f(x_1, x_2, \ldots, x_n)$.

For example, consider the addition over a subset of $\mathbb{N}_{\geq 0}$ of size polynomial in n, i.e., $S = [\mathrm{poly}(n)] \cup \{\mathrm{NaN}\}$ and

$$\oplus : S \times S \to S, (x, y) \mapsto \begin{cases} x + y & \text{if } x, y \neq \mathrm{NaN} \text{ and } x + y \leq \max_{s \in S \setminus \{\mathrm{NaN}\}}\{s\} \\ \mathrm{NaN} & \text{else} \end{cases}.$$

Here, NaN (*not a number*) denotes a number larger than $\max_{s \in S \setminus \{\mathrm{NaN}\}}\{s\}$. Since n is assumed to be known, the nodes are able to obtain the average of all inputs by solving a summation problem.

Finally, the following problem is essentially the all-to-all token dissemination problem that allows the entire reconstruction of each input.

Problem 5.3 (Noncontinuous Dissemination)**.** Let S be a structure representable by $\mathrm{O}(\log n)$ bits. Each node i in the network receives an element $x_i \in S$ called token as input. All nodes of the network have to output $x_1, x_2, \dots, x_n$.

In addition to these noncontinuous problems, there are continuous variants of these problems, in which f has not to be computed only once, but several times for different inputs.

Problem 5.4 (Continuous Extremum)**.** Define f and $(S, \oplus)$ as in the noncontinuous extremum problem. In each round r, each node i in the network receives an element $x_{i,r} \in S$ as input. For a subset of rounds $R \subseteq \mathbb{N}_{\geq 0}$ (defined by the algorithm) and each $r' \in R$, all nodes have to output $f(x_{1,r'}, x_{2,r'}, \dots, x_{n,r'})$.

Problem 5.5 (Continuous Summation)**.** Define f and $(S, \oplus)$ as in the noncontinuous summation problem. In each round r, each node i in the network receives an element $x_{i,r} \in S$ as input. For a subset of rounds $R \subseteq \mathbb{N}_{\geq 0}$ (defined by the algorithm) and each $r' \in R$, all nodes have to output $f(x_{1,r'}, x_{2,r'}, \dots, x_{n,r'})$.

Problem 5.6 (Continuous Dissemination)**.** Define f and S as in the noncontinuous dissemination problem. In each round r, each node i in the network receives an element $x_{i,r} \in S$ as input. For a subset of rounds $R \subseteq \mathbb{N}_{\geq 0}$ (defined by the algorithm) and each $r' \in R$, all nodes have to output $f(x_{1,r'}, x_{2,r'}, \dots, x_{n,r'})$.

Note that solving these problems for one round, in general, requires more than just one round. Although it is possible to produce a result for each round $r \in \mathbb{N}_{\geq 0}$, it could take longer and longer: Let T be the number of rounds required to produce one output. Then, it is possible to output the result of round r in round $r \cdot T$ by running the algorithm for each round. Since the algorithms studied in this chapter are intended to run for a long time, this is not a feasible approach and the number of rounds to produce one output shall not depend on the round, once the computation has started. Instead, the algorithms should drop some rounds and produce outputs without this dependence.

Intuitively, a good algorithm for continuous problems produces as many results as possible and requires few rounds per output. This is captured by the following two metrics which are used to evaluate the algorithms in this chapter.

Definition 5.7 (Output Rate)**.** The *output rate* of an algorithm is defined as

$$\lim_{r\to\infty} \frac{\#\text{results up to round } r}{r}.$$

Definition 5.8 (Delay)**.** The *delay* of an algorithm is defined as the maximum number of rounds between the round, in which inputs arrive and the round the function of these inputs is output by all nodes.

These definitions of the output rate and the delay allow the algorithms to perform an initial setup phase without changing the output rate. For example, if n is not known, then the nodes are able to obtain the total count in $\mathrm{O}(n^2)$ rounds before they start to solve a continuous problem.

5.2 Related Work

For static networks, it is known that many problems such as computing the maximum, sum, parity, or majority can be solved in linear time in a graph by computing a spanning tree at first (see, e.g., Awerbuch [Awe87]). More specifically, if D is the diameter of the graph, all these functions can be computed in $\mathrm{O}(D)$ rounds. Beyond that, more complicated problems have been studied, e.g., selection problems [KLW07] or the problem of computing the mode (most frequent element) [KLS08].

For dynamic networks, Cornejo et al. [CGN12] studied a different aggregation problem, in which tokens have to be gathered at a minimum number of nodes. On the one hand, they proved that there is no algorithm with a good competitive ratio compared to an optimal offline algorithm. On the other hand, under the assumption that every node interacts with at least a p-fraction of the nodes, they presented an algorithm that aggregates the tokens to a logarithmic number of nodes with high probability.

Mosk-Aoyama and Shah [MS06] showed how so-called separable functions can be approximated with a gossiping algorithm based on exponential random variables. Their techniques can also be applied to dynamic networks as Kuhn et al. [KLO10] showed for approximate counting.

A main building block of this chapter is the construction of maximal independent sets (MIS). The distributed algorithm by Luby [Lub86] computes an MIS in expected $O(\log n)$ rounds. It can also be shown that the number of rounds is $O(\log n)$ with high probability [Kar94; CD97].

The best known distributed deterministic algorithm by Panconesi and Srinivasan [PS92] computes an MIS in $2^{O(\sqrt{\log n})}$ rounds. In growth-bounded graphs, Schneider and Wattenhofer [SW08] show how to deterministically create an MIS in $O(\log^* n)$ communication rounds. This is asymptotically optimal as Linial [Lin92] gave a corresponding $\Omega(\log^* n)$ lower bound.

5.3 Static Networks

Before proceeding to T-stable dynamic networks, it is first discussed how the problems can be solved in static networks and which delays and output rates can be achieved.

5.3.1 Noncontinuous Problems

Solving all three noncontinuous problems requires a linear number of rounds in static networks. For the sake of a simpler presentation of the algorithms for T-stable dynamic networks, the algorithms for these problems in static networks are discussed in detail.

Theorem 5.9 (Noncontinuous Extremum, Summation, and Dissemination in Static Networks)**.** *In static networks, the extremum, the summation, and the dissemination problem can be solved in* $O(n)$ *rounds.*

Proof. The extremum problem can be solved in $n-1$ rounds: Each node i initially broadcasts its input x_i. In every other round up to round $n-1$, each node i takes all its incoming messages $m_1, \dots, m_l$ and broadcasts $\bigoplus_{j=1}^{l} m_j \oplus x_i$. In round $n-1$, the sum $\bigoplus_{j=1}^{l} m_j \oplus x_i$ is equal to $f(x_1, \dots, x_n)$ since all inputs must be contained in this sum and multiplicities cancel out due to the associativity, commutativity, and idempotence of the binary operation of the semigroup. See Algorithm 5.1 for a pseudocode description. As described later, it can also be used in 1-stable dynamic networks.

Algorithm 5.1: Noncontinuous Extremum in Static Networks (and 1-Stable Dynamic Networks)

$x \leftarrow x_i$
for $r \leftarrow 1, \dots, n-1$ **do**
  send x
  $m_1, \dots, m_l \leftarrow$ values received from neighbors
  $x \leftarrow \bigoplus_{j=1}^{l} m_j \oplus x$
return x

To solve the summation problem, note that the node with the smallest unique ID can be found in $n-1$ rounds (this is an extremum problem). Then, in further $n-1$ rounds, a breadth first search tree that is rooted at the node with the smallest unique ID can be built. Along this tree, starting from the leaves up to the root, the inputs can be summed up and finally the result can be broadcasted back from the root to all nodes of the tree. It is thereby guaranteed that each summand is considered exactly once in the total sum. Therefore, $O(n)$ rounds are required to solve the summation problem. See Algorithm 5.2 for a pseudocode description.

Algorithm 5.2: Noncontinuous Summation in Static Networks

---

find the smallest unique ID (see Algorithm 5.2)
build a BFS tree rooted at the node having the smallest unique ID
$m_1, \dots, m_l \leftarrow$ values received from all l children (wait if necessary)
$s \leftarrow \bigoplus_{j=1}^{l} m_j \oplus x_i$
send s
broadcast s from the root to all nodes
return s

Finally, to solve the dissemination problem, a breadth first search tree can be built, similar to the one for the summation problem. Then, in each round, each of the tree's nodes sends a token, it has not yet sent upwards, to the root of the tree such that the root holds all tokens after $O(n)$ rounds. Subsequently, the tokens are sent one after the other from the root to the leaves, i.e., one token received from the parent is sent to the children directly in the following round. Thus, $O(n)$ rounds are required in total. See Algorithm 5.2 for a pseudocode description.

Algorithm 5.3: Noncontinuous Dissemination in Static Networks

find the smallest unique ID (see Algorithm 5.2)
build a BFS tree rooted at the node having the smallest unique ID
send own token x_i
while ∃token t received from children that has not yet been sent
  send t
broadcast all tokens from the root to all nodes and return them

This yields the claim that all three noncontinuous problems can be solved in a linear number of rounds. □

Note that the algorithm that solves the dissemination problem could have been used to solve the extremum and the summation problem within the same asymptotic running time. However, the aforementioned algorithms have been presented to simplify the presentation of the continuous variants in T-stable dynamic networks.

5.3.2 Continuous Problems

The continuous variants of the extremum and summation problem can be solved with linear delay and constant output rate.

Theorem 5.10 (Continuous Extremum and Summation in Static Networks)**.** *In static networks, the continuous extremum problem and the continuous summation problem can be solved with delay* $O(n)$ *and output rate* $\Omega(1)$*. The delays and the output rates are asymptotically optimal.*

Proof. To continuously solve both the extremum and the summation problem, a breadth first search tree is built, as before, and a pipelining technique is applied. The leaves of the tree start sending their inputs from the first round upwards. In the next round, the leaves start sending their inputs from the second round upwards and so on until round n. The nodes with maximum distance l to a leave within the tree in round r sum up the incoming messages from the level below and add their inputs from round $r - l$ if $r - l > 0$. Then, after at most $2n$ rounds, the results for round 1 to n are available at the root node and can be sent one after the other from the root to the leaves as the tokens in the dissemination problem. See Algorithm 5.2 for a pseudocode description.

Algorithm 5.4: Continuous Extremum and Summation in Static Networks

find the smallest unique ID (see Algorithm 5.2)
build a BFS tree rooted at the node having the smallest unique ID
let $m_{j,k}$ be the k^{th} value received from the j^{th} child
while true
  $r \leftarrow$ current round number
  $c \leftarrow 1$
  for $r' \leftarrow 1, \dots, 2n$ **do**
    receive values
    if received $m_{1,c}, \dots, m_{l,c}$ from all l children
      $s_{r+c} \leftarrow \bigoplus_{j=1}^{l} m_{j,c} \oplus x_{i,r+c}$
      send s_{r+c}
      $c \leftarrow c + 1$
    else
      send nothing
  broadcast $s_{r+1}, \dots, s_{r+n}$ from the root to all nodes and return them

This gives n outputs in $\mathrm{O}(n)$ rounds. Since the best possible output rate is 1 and the delay cannot be better than the diameter of the network (which could be n), this yields the claim. □

The continuous dissemination problem is different. It can still be solved in a linear number of rounds, but the output rate is $\Theta(\frac{1}{n})$.

Theorem 5.11 (Continuous Dissemination in Static Networks)**.** *In static networks, the continuous dissemination problem can be solved with delay* $\mathrm{O}(n)$ *and output rate* $\Omega(\frac{1}{n})$*. The delay and the output rate are asymptotically optimal.*

Proof. For the dissemination problem, it is impossible to achieve better delays and output rates than those obtained by just solving the noncontinuous version over and over again (cf. Algorithm 5.5).

Algorithm 5.5: Continuous Dissemination in Static Networks

while true
  execute Algorithm 5.3 with the inputs of the current round

The delay is bounded by the diameter of the network. For the output rate, consider $|S| = n$, a line of n nodes and the edge e with $\lceil \frac{n}{2} \rceil$ nodes on the left and $\lfloor \frac{n}{2} \rfloor$ nodes on the right. If the output rate was $\omega(\frac{1}{n})$, then $\omega(\frac{r}{n})$ outputs must have been computed up to round r. Clearly, $\mathrm{O}(r \cdot \log n)$ bits could have passed e from the left to the right in r rounds. These bits separate up to $n^{\mathrm{O}(r)}$ instances. However, there are $\binom{|S|}{n/2}^{\omega(r/n)} = n^{\omega(r)}$ possibilities to choose the tokens on the left side. Hence, at least one output must be wrong. □

5.4 T-Stable Dynamic Networks

First, it is shown how a T-stable dynamic network can be partitioned into so-called patches. This partitioning allows to aggregate data within these patches and it is used to solve the summation problem, the continuous extremum, and the continuous summation problem.

5.4.1 Graph Patching Technique

The following patching idea is adapted from Haeupler and Karger [HK11].

Definition 5.12 ((S, D)-Patch, (S, D)-Patching)**.** An (S, D)-*patch* of a graph $G = (V, E)$ is a rooted tree in G that spans at least S nodes and has depth at most D. An (S, D)-*patching* of a graph is a set of (S, D)-patches such that the sets of the nodes of all (S, D)-patches give a disjoint partition of V.

A $\left(\frac{D}{2}, D\right)$-patching of G can be distributedly computed by first finding a set of nodes in G that form a maximal independent set (MIS) in G^D, i.e., the D^{th} power of G, and then computing breadth first search trees from all nodes in the MIS, in which each node not in the MIS is assigned to one of its closest MIS nodes (cf. Algorithm 5.6 and Figure 5.1 for an exemplary $\left(\frac{D}{2}, D\right)$-patching computed by this algorithm with $D = 4$). Thereby, the diameter of each patch is at most D and each patch contains at least $\frac{D}{2}$ nodes, since each node has an MIS node within distance D and the distance between two MIS nodes in G is at least $D + 1$.

Algorithm 5.6: $\left(\frac{D}{2}, D\right)$-Patching in Static Networks

```
1  compute an MIS in G^D
2  partition the graph into trees by performing one BFS from all nodes in the MIS
```

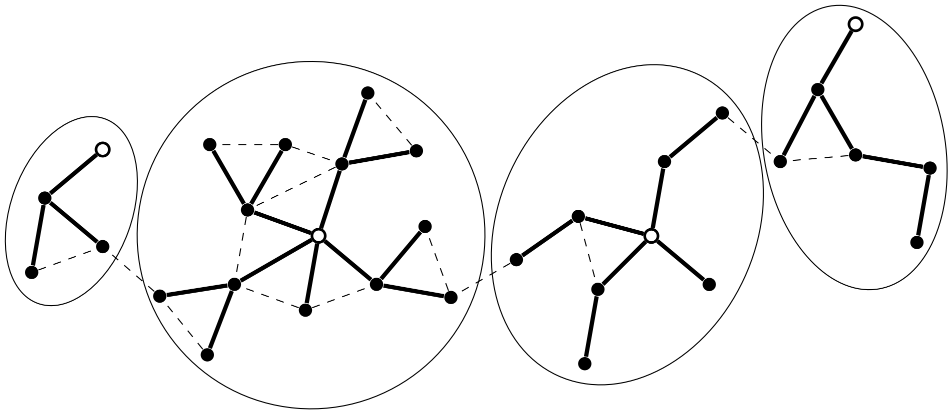

Figure 5.1: Example for a $\left(\frac{D}{2}, D\right)$-patching computed by Algorithm 5.6 with $D = 4$: Consider the undirected graph given by all dashed and thick edges. Each ellipse marks a patch, in which the white node is the root node and the thick edges are the edges of a tree. The white nodes form a maximal independent set in G^4. Each patch contains at least 2 nodes and the tree's depth in a patch is at most 4.

Existing distributed MIS algorithms can be adapted to compute an MIS in G^D. For this, MIS algorithms running in G^D can be simulated in G by relaying messages up to D hops, i.e., a message sent over one edge in G^D corresponds to a path of length at most D in G. If an MIS algorithm can be modified such that the congestion caused by overlapping paths is not too high, then the MIS algorithm is slowed down by a factor of at most D. Such modifications are possible for many existing MIS algorithms because nodes usually try to rule out other nodes and it is sufficient to relay one message only. Let $\mathrm{MIS}(n)$ be the number of rounds required by some algorithm to compute an MIS in a graph with n nodes and assume this algorithm can be modified as described. Then, a $\left(\frac{D}{2}, D\right)$-patching can be computed in $\mathrm{O}(\mathrm{MIS}(n) \cdot D)$ rounds. This yields the following known results.

Theorem 5.13 (Deterministic $\left(\frac{D}{2}, D\right)$-Patching)**.** *A graph G can be partitioned into $\left(\frac{D}{2}, D\right)$-patches in $\mathrm{O}\left(2^{\mathrm{O}(\sqrt{\log n})} \cdot D\right)$ rounds with Panconesi and Srinivasan's deterministic MIS algorithm. [HK11; PS92]*

Theorem 5.14 (Randomized $\left(\frac{D}{2}, D\right)$-Patching)**.** *A graph G can be partitioned into $\left(\frac{D}{2}, D\right)$-patches in $\mathrm{O}(\log(n) \cdot D)$ rounds with high probability with Luby's randomized MIS algorithm. [HK11; Lub86; Kar94; CD97]*

In growth-bounded graphs, a patching can be computed faster which helps speeding up the process in geometric dynamic networks.

Theorem 5.15 (Deterministic $\left(\frac{D}{2}, D\right)$-Patching in Growth-Bounded Graphs)**.** *A growth-bounded graph G can be partitioned into* $\left(\frac{D}{2}, D\right)$*-patches in* $O\left(\log^*(n) \cdot D\right)$ *rounds with Schneider and Wattenhofer's deterministic MIS algorithm.*

Proof. The algorithm by Schneider and Wattenhofer [SW08] can be modified such that it can be applied here: In each competition and whenever the states are updated, each competitor is interested in the competitor u in its neighborhood that has the minimum result r_u^{j-1} among all its neighbors. In addition, the nodes only need to know whether there exists a competitor, a ruler, or a dominator in their neighborhood. Therefore, each node only needs to flood the minimum result of a competitor, i.e., whether there exists a ruler or a dominator for D rounds. □

5.4.2 Noncontinuous Extremum

Despite the presence of an adaptive adversary, the extremum problem can be solved without the need for a graph patching. This is a tight result, since—even in static networks—the extremum problem cannot be solved faster.

Theorem 5.16 (Noncontinuous Extremum in T-Stable Dynamic Networks)**.** *In* 1*-stable /* 1*-interval connected dynamic networks, the noncontinuous extremum problem can be solved in* $O(n)$ *rounds.*

Proof. The algorithm that solves the extremum problem in dynamic networks is the same as the algorithm used for static networks (cf. Algorithm 5.1): Each node i initially broadcasts its input x_i. In every other round up to round $n-1$, each node i takes all its incoming messages $m_1, \ldots, m_l$ and broadcasts $\bigoplus_{j=1}^{l} m_j \oplus x_i$.

In round $n-1$, the sum $\bigoplus_{j=1}^{l} m_j \oplus x_i$ is equal to $f(x_1, \ldots, x_n)$ because it contains all inputs (each node causally influenced every other node after $n-1$ rounds [KLO10]) and multiplicities cancel out due to the associativity, commutativity, and idempotence of the semigroup's binary operation. □

5.4.3 Noncontinuous Summation

If the dynamic networks are stable enough, then the noncontinuous summation problem can be solved faster than the noncontinuous dissemination problem by applying the patching technique.

Theorem 5.17 (Noncontinuous Summation in T-Stable Dynamic Networks)**.** *In T-stable dynamic networks with $T \geq c \cdot \mathrm{MIS}(n)$ for a sufficiently large constant c, the noncontinuous summation problem can be solved in*

- $\mathrm{O}\left(\frac{n^2 \cdot \mathrm{MIS}(n)}{T^2}\right)$ *rounds if* $T = \mathrm{O}\left(\sqrt{n \cdot \mathrm{MIS}(n)}\right)$ *and*
- $\mathrm{O}(n)$ *rounds if* $T = \Omega\left(\sqrt{n \cdot \mathrm{MIS}(n)}\right)$.

Proof. Consider the following algorithm.

Algorithm 5.7: Noncontinuous Summation in T-Stable Dynamic Networks

choose $D = \Theta\left(\frac{T}{\mathrm{MIS}(n)}\right)$ appropriately and compute a $\left(\frac{D}{2}, D\right)$−patching
in each patch, compute the sum of all inputs of the nodes in the patch
disseminate all partial sums of the patches to all nodes and sum them up

If c is large enough and D is chosen properly, the first and the second step can be conducted in at most T rounds. Since each patch contains at least $\frac{D}{2}$ nodes, there are $\mathrm{O}\left(\frac{n}{D}\right) = \mathrm{O}\left(\frac{n \cdot \mathrm{MIS}(n)}{T}\right)$ partial sums left in step three. To disseminate these, the token dissemination algorithm by Kuhn et al. [KLO10] for T-stable connected dynamic networks can be used (cf. Section 2.4.1). Thus, the summation problem can be solved in $\mathrm{O}\left(\frac{n^2 \cdot \mathrm{MIS}(n)}{T^2} + n\right)$ rounds. □

If the deterministic MIS algorithm by Panconesi and Srinivasan [PS92] is used for computing the $\left(\frac{D}{2}, D\right)$-patchings, this yields the following result.

Corollary 5.18 (Noncontinuous Summation in T-Stable Dynamic Networks, Deterministic Algorithm)**.** *In T-stable dynamic networks with $T \geq 2^{c \cdot \sqrt{\log n}}$ for a sufficiently large constant c, the summation problem can be solved in*

- $\mathrm{O}\left(\frac{n^2}{T^2} \cdot 2^{c \cdot \sqrt{\log n}}\right)$ *rounds if* $T = \mathrm{O}\left(\sqrt{n} \cdot \sqrt{2}^{c \cdot \sqrt{\log n}}\right)$ *and*
- $\mathrm{O}(n)$ *rounds if* $T = \Omega\left(\sqrt{n} \cdot \sqrt{2}^{c \cdot \sqrt{\log n}}\right)$.

Randomization allows to speed up this computation if the randomized MIS algorithm by Luby [Lub86] is used to compute the $\left(\frac{D}{2}, D\right)$-patching and the randomized network coding algorithm by Haeupler and Karger [HK11] is used for dissemination.

Corollary 5.19 (Noncontinuous Summation in T-Stable Dynamic Networks, Randomized Algorithm)**.** *Let $L = \mathrm{O}(\log n)$ be the number of rounds Luby's algorithm needs to compute a maximal independent set with high probability. Then, in T-stable dynamic networks with $T \geq L$, the summation problem can be solved within the number of rounds listed in Table 5.4a.*

Proof. Choose $D = \Theta\left(\frac{T}{L}\right)$ such that $\leq T$ rounds are needed to compute a $\left(\frac{D}{2}, D\right)$-patching and to sum up all values in each patch. Now, the randomized network coding algorithm by Haeupler and Karger [HK11] can be used for dissemination. It needs

$$\mathrm{O}\left(\min\left\{\frac{nk}{T^2} + T^2 n \log^2 n, \frac{nk \log n}{T^2} + Tn \log^2 n, \frac{n^2 \log n}{T^2} + n \log n\right\}\right)$$

rounds to disseminate k tokens with high probability. For different ranges of T and $k = \mathrm{O}\left(\frac{n \cdot L}{T}\right)$, the following number of rounds are needed with high probability.

(1) $\mathrm{O}\left(\frac{n^2 \log n}{T^3}\right)$ if $T = \mathrm{O}\left(n^{1/5} \log^{-1/5} n\right)$

(2) $\mathrm{O}\left(T^2 n \log^2 n\right)$ if $\Omega\left(n^{1/5} \log^{-1/5} n\right) = T = \mathrm{O}\left(n^{1/5}\right)$

(3) $\mathrm{O}\left(\frac{n^2 \log^2 n}{T^3}\right)$ if $\Omega\left(n^{1/5}\right) = T = \mathrm{O}\left(n^{1/4}\right)$

(4) $\mathrm{O}\left(Tn \log^2 n\right)$ if $\Omega\left(n^{1/4}\right) = T = \mathrm{O}\left(n^{1/3} \log^{-1/3} n\right)$

(5) $\mathrm{O}\left(\frac{n^2 \log n}{T^2}\right)$ if $\Omega\left(n^{1/3} \log^{-1/3} n\right) = T = \mathrm{O}\left(n^{1/2}\right)$

(6) $\mathrm{O}(n \log n)$ if $\Omega\left(n^{1/2}\right) = T$

Note that the number of rounds in the second and fourth range increases with T. However, a T-stable dynamic network is also $\frac{T}{l}$-stable for any $l > 1$. Therefore, T can be replaced by the lower bound of the range.

(1) $\mathrm{O}\left(\frac{n^2 \log n}{T^3}\right)$ if $T = \mathrm{O}\left(n^{1/5} \log^{-1/5} n\right)$

(2) $O\left(n^{7/5}\log^{8/5} n\right)$ if $\Omega\left(n^{1/5}\log^{-1/5} n\right) = T = O\left(n^{1/5}\log n^{2/15}\right)$

(3) $O\left(\frac{n^2\log^2 n}{T^3}\right)$ if $\Omega\left(n^{1/5}\log n^{2/15}\right) = T = O\left(n^{1/4}\right)$

(4) $O\left(n^{5/4}\log^2 n\right)$ if $\Omega\left(n^{1/4}\right) = T = O\left(n^{3/8}\log^{-1/2} n\right)$

(5) $O\left(\frac{n^2\log n}{T^2}\right)$ if $\Omega\left(n^{3/8}\log^{-1/2} n\right) = T = O\left(n^{1/2}\right)$

(6) $O(n\log n)$ if $\Omega\left(n^{1/2}\right) = T$

This gives the results for the noncontinuous summation problem listed in Table 5.4a. □

5.4.4 Continuous Extremum

The continuous extremum problem can be solved with a higher output rate than the one obtained by just executing the algorithm for the noncontinuous problem over and over again. For this, the $\left(\frac{D}{2}, D\right)$-patching technique is applied.

Theorem 5.20 (Continuous Extremum in T-Stable Dynamic Networks)**.** *In T-stable dynamic networks with $T \geq c \cdot \mathrm{MIS}(n)$ for a sufficiently large constant c, the continuous extremum problem can be solved with delay $O(n \cdot \mathrm{MIS}(n))$ and output rate $\Omega\left(\frac{T}{n\cdot\mathrm{MIS}(n)^2}\right)$.*

Proof. Consider the following algorithm. For an appropriately chosen $D = \Theta\left(\frac{T}{\mathrm{MIS}(n)}\right)$, it produces D outputs in $O(n \cdot \mathrm{MIS}(n))$ rounds.

(1) Each node $i \in V$ initializes $y_{i,r,0}$ with $x_{i,r}$ for $r = 1, \dots, D$.

(2) For $j = 1, \dots, \frac{2n}{D}$ phases of T rounds do:

   (a) Compute a $\left(\frac{D}{2}, D\right)$-patching.

   (b) Each node i in each patch P computes $y_{i,r,j}$ as the sum of $y_{i',r,j-1}$ for all nodes i' from P and all adjacent patches of P for $r = 1, \dots, D$.

(3) Each node $i \in V$ returns $y_{i,r,\frac{2n}{D}}$ for $r = 1, \dots, D$.

If c is large enough and D is chosen properly, (a) and (b) can be done in a stable phase of T rounds. Consider any input $x_{i,r}$ with $r \in \{1, \dots, D\}$. A patch

P is referred to as *knowing* $x_{i,r}$ if and only if $x_{i,r}$ is contained in any $y_{i',r,j}$ for $i' \in P$. If there is a patch P that does not know $x_{i,r}$ at the beginning of a phase, then there is a patch P^* that does not know $x_{i,r}$ at the beginning of the phase but knows $x_{i,r}$ at the end of the phase. Thus, at least $\frac{D}{2}$ nodes learn about $x_{i,r}$ in each phase until all nodes get to know $x_{i,r}$. Furthermore, after $\frac{2n}{D}$ phases, all inputs $x_{i,r}$ are contained in all $y_{i',r,\frac{2n}{D}}$ for $r \in \{1, \dots, D\}$. Therefore, after $\frac{2n}{D} \cdot T = \mathrm{O}(n \cdot \mathrm{MIS}(n))$ rounds, D outputs have been generated yielding the claimed delay and output rate. See Algorithm 5.8 for a pseudocode description of the continuous algorithm.

Algorithm 5.8: Continuous Extremum in T-Stable Dynamic Networks

```
choose D = Θ(T / MIS(n)) appropriately
while true
  r ← current round number
  for r' ← 1, …, D do
    y_{i,r+r',0} ← x_{i,r+r'}
  for j ← 1, …, 2n/D phases of T rounds do
    compute a (D/2, D)−patching
    compute y_{i,r+r',j} ← ⊕_{i'∈P∪N(P)} y_{i',r+r',j−1} for r' ← 1, …, D
  return y_{i,r'+1,2n/D}, …, y_{i,r'+D,2n/D}
```

□

Again, if the deterministic MIS algorithm by Panconesi and Srinivasan [PS92] is used for computing the $\left(\frac{D}{2}, D\right)$-patchings, this yields the following result.

Corollary 5.21 (Continuous Extremum in T-Stable Dynamic Networks, Deterministic Algorithm)**.** *In T-stable dynamic networks with* $T \geq 2^{c \cdot \sqrt{\log n}}$ *for a sufficiently large constant c, the continuous extremum problem can be solved with delay* $\mathrm{O}\left(n \cdot 2^{c \cdot \sqrt{\log n}}\right)$ *and output rate* $\Omega\left(\frac{T}{n \cdot 2^{c \cdot \sqrt{\log n}}}\right)$.

Again, randomization allows to speed up this computation.

Corollary 5.22 (Continuous Extremum in T-Stable Dynamic Networks, Randomized Algorithm)**.** *Let* $L = \mathrm{O}(\log n)$ *be the number of rounds Luby's algorithm needs to compute a maximal independent set with high probability. Then, in T-stable dynamic networks with* $T \geq L$*, the continuous extremum problem can be solved with output rate* $\Omega\left(\frac{T}{n \log n}\right)$ *and delay* $\mathrm{O}(n \log n)$ *with high probability.*

Proof. Choose $D = \Theta\left(\frac{T}{L}\right)$ such that $\leq T$ rounds are needed to compute a $\left(\frac{D}{2}, D\right)$-patching and to do the computations in the patches as in the proof of Theorem 5.20. If this is repeated $\frac{2n}{D}$ times, then, with high probability, there are still at least $\frac{2n}{D}$ valid $\left(\frac{D}{2}, D\right)$-patchings. Therefore, with high probability, after $\frac{2n}{D} \cdot T = \mathrm{O}(n \log n)$ rounds, D outputs can be generated, producing the claimed delay and output rate. □

5.4.5 Continuous Summation

Finally, the continuous summation problem can be solved by first computing partial sums for several rounds within a patch and disseminating these partial sums afterwards.

Theorem 5.23 (Continuous Summation in T-Stable Dynamic Networks)**.** *In T-stable dynamic networks with $T \geq c \cdot \mathrm{MIS}(n)$ for a sufficiently large constant c, the continuous summation problem can be solved with delay*

- $\mathrm{O}\left(\frac{n^2 \cdot \mathrm{MIS}(n)^2}{T^{3/2}}\right)$ *if* $T = \mathrm{O}\left(n^{2/3} \cdot \mathrm{MIS}(n)^{2/3}\right)$ *and*
- $\mathrm{O}(n \cdot \mathrm{MIS}(n))$ *if* $T = \Omega\left(n^{2/3} \cdot \mathrm{MIS}(n)^{2/3}\right)$

and output rate

- $\Omega\left(\frac{T^{5/2}}{n^2 \cdot \mathrm{MIS}(n)^3}\right)$ *if* $T = \mathrm{O}\left(n^{2/3} \cdot \mathrm{MIS}(n)^{2/3}\right)$ *and*
- $\Omega\left(\frac{T}{n \cdot \mathrm{MIS}(n)^2}\right)$ *if* $T = \Omega\left(n^{2/3} \cdot \mathrm{MIS}(n)^{2/3}\right)$.

Proof. Consider the following algorithm.

Algorithm 5.9: Continuous Summation in T-Stable Dynamic Networks

1 choose $D = \Theta\left(\frac{T}{\mathrm{MIS}(n)}\right)$ appropriately
2 **while true**
3   compute a $\left(\frac{D}{2}, D\right)$−patching
4   compute $\frac{D}{2}$ sums of $\frac{D}{2}$ rounds of all inputs within each patch
5   disseminate all partial sums of the patches to all nodes and sum them up
6   return the results

If c is large enough and D is chosen properly, then the third and the fourth step can be done in at most T rounds. Since each patch has a size of at least $\frac{D}{2}$ (if

Algorithm 5.6 is used), there are at most n partial sums left. Now, the deterministic variant of the network coding algorithm by Haeupler and Karger [HK11] can be applied. It is able to disseminate $k \leq n$ tokens in

$$O\left(\left(\frac{n \cdot \mathrm{MIS}(n)}{\sqrt{T}} \cdot \min\left\{k \cdot \sqrt{\log n}, \frac{n}{T}\right\} + n\right) \cdot \mathrm{MIS}(n)\right)$$

rounds. Note that it is faster than the token dissemination algorithm by Kuhn et al. [KLO10] for $T \geq \mathrm{MIS}(n)$ and $k = n$.

Therefore, it is possible to disseminate all these up to n partial sums in $O\left(\left(\frac{n^2 \cdot \mathrm{MIS}(n)}{T^{3/2}} + n\right) \cdot \mathrm{MIS}(n)\right)$ rounds. If $T = O(n^{2/3} \cdot \mathrm{MIS}(n)^{2/3})$, then $\frac{D}{2}$ outputs can be generated in $O\left(T + \frac{n^2 \cdot \mathrm{MIS}(n)^2}{T^{3/2}}\right)$ rounds and an output rate of $\Omega\left(\frac{T^{5/2}}{n^2 \cdot \mathrm{MIS}(n)^3}\right)$ is achieved. If $T = \Omega(n^{2/3} \cdot \mathrm{MIS}(n)^{2/3})$, then $\frac{D}{2}$ outputs can be generated in $O(n \cdot \mathrm{MIS}(n))$ rounds and an output rate of $\Omega\left(\frac{T}{n \cdot \mathrm{MIS}(n)^2}\right)$ is achieved. □

Again, if the deterministic MIS algorithm by Panconesi and Srinivasan [PS92] is used for computing the $(\frac{D}{2}, D)$-patchings, this yields the following improved results.

Corollary 5.24 (Continuous Summation in T-Stable Dynamic Networks, Deterministic Algorithm)**.** *In T-stable dynamic networks with* $T \geq 2^{c \cdot \sqrt{\log n}}$ *for a sufficiently large constant c, the continuous summation problem can be solved with delay*

- $O\left(\frac{n^2 \cdot 2^{2c \cdot \sqrt{\log n}}}{T^{3/2}}\right)$ *if* $T = O\left(n^{2/3} \cdot 2^{c \cdot \frac{2}{3} \cdot \sqrt{\log n}}\right)$ *and*
- $O\left(n \cdot 2^{c \cdot \sqrt{\log n}}\right)$ *if* $T = \Omega\left(n^{2/3} \cdot 2^{c \cdot \frac{2}{3} \cdot \sqrt{\log n}}\right)$

and output rate

- $\Omega\left(\frac{T^{5/2}}{n^2 \cdot 2^{3c \cdot \sqrt{\log n}}}\right)$ *if* $T = O\left(n^{2/3} \cdot 2^{c \cdot \frac{2}{3} \cdot \sqrt{\log n}}\right)$ *and*
- $\Omega\left(\frac{T}{2^{c \cdot \sqrt{\log n}}}\right)$ *if* $T = \Omega\left(n^{2/3} \cdot 2^{c \cdot \frac{2}{3} \cdot \sqrt{\log n}}\right)$.

Again, randomization allows to speed up this computation if the randomized MIS algorithm by Luby [Lub86] is used to compute the $(\frac{D}{2}, D)$-patching and the randomized network coding algorithm by Haeupler and Karger [HK11] is used for dissemination.

Corollary 5.25 (Continuous Summation in T-Stable Dynamic Networks, Randomized Algorithm)**.** *Let $L = \mathrm{O}(\log n)$ be the number of rounds Luby's algorithm needs to compute a maximal independent set with high probability. Then, in T-stable dynamic networks with $T \geq L$, the continuous summation problem can be solved with the output rates and delays as listed in Table 5.4b.*

Proof. Choose $D = \Theta\left(\frac{T}{L}\right)$ such that $\leq T$ rounds are needed to compute a $\left(\frac{D}{2}, D\right)$-patching and to do the computations in the patch as in the proof of Theorem 5.23. Now, the randomized network coding algorithm by Haeupler and Karger [HK11] can be used for dissemination. It requires

$$\mathrm{O}\left(\min\left\{\frac{nk}{T^2} + T^2 n \log^2 n, \frac{nk \log n}{T^2} + Tn \log^2 n, \frac{n^2 \log n}{T^2} + n \log n\right\}\right)$$

rounds to disseminate k tokens with high probability. For different ranges of T and $k = n$, the following number of rounds are needed with high probability.

(1) $\mathrm{O}\left(\frac{n^2}{T^2}\right)$ if $T = \mathrm{O}\left(n^{1/4} \log^{-1/2} n\right)$

(2) $\mathrm{O}\left(T^2 n \log^2 n\right)$ if $\Omega\left(n^{1/4} \log^{-1/2} n\right) = T = \mathrm{O}\left(n^{1/4} \log^{-1/4} n\right)$

(3) $\mathrm{O}\left(\frac{n^2 \log n}{T^2}\right)$ if $\Omega\left(n^{1/4} \log^{-1/4} n\right) = T = \mathrm{O}\left(n^{1/2}\right)$

(4) $\mathrm{O}(n \log n)$ if $\Omega\left(n^{1/2}\right) = T$

Note that the number of rounds in the second range increases with T. However, a T-stable dynamic network is also $\frac{T}{l}$-stable for any $l > 1$. Therefore, T can be replaced by the lower bound of the range.

(1) $\mathrm{O}\left(\frac{n^2}{T^2}\right)$ if $T = \mathrm{O}\left(n^{1/4} \log^{-1/2} n\right)$

(2) $\mathrm{O}\left(n^{3/2} \log n\right)$ if $\Omega\left(n^{1/4} \log^{-1/2} n\right) = T = \mathrm{O}\left(n^{1/4}\right)$

(3) $\mathrm{O}\left(\frac{n^2 \log n}{T^2}\right)$ if $\Omega\left(n^{1/4}\right) = T = \mathrm{O}\left(n^{1/2}\right)$

(4) $\mathrm{O}(n \log n)$ if $\Omega\left(n^{1/2}\right) = T$

This gives the results for the continuous summation problem in Table 5.4b. □

Table 5.3: Summation in T-Stable Dynamic Networks with $T \geq L$.

(a) Noncontinuous Summation.

Running Time	**Range for T**
$O\left(\frac{n^2 \log n}{T^3}\right)$ w.h.p.	if $L \leq T = O\left(n^{1/5} \log^{-1/5} n\right)$
$O\left(n^{7/5} \log^{8/5} n\right)$ w.h.p.	if $\Omega\left(n^{1/5} \log^{-1/5} n\right) = T = O\left(n^{1/5} \log n^{2/15}\right)$
$O\left(\frac{n^2 \log^2 n}{T^3}\right)$ w.h.p.	if $\Omega\left(n^{1/5} \log n^{2/15}\right) = T = O\left(n^{1/4}\right)$
$O\left(n^{5/4} \log^2 n\right)$ w.h.p.	if $\Omega\left(n^{1/4}\right) = T = O\left(n^{3/8} \log^{-1/2} n\right)$
$O\left(\frac{n^2 \log n}{T^2}\right)$ w.h.p.	if $\Omega\left(n^{3/8} \log^{-1/2} n\right) = T = O\left(n^{1/2}\right)$
$O(n \log n)$ w.h.p.	if $\Omega\left(n^{1/2}\right) = T \leq n$

(b) Continuous Summation.

Delay	**Output Rate**	**Range for T**
$O\left(\frac{n^2}{T^2}\right)$ w.h.p.	$\Omega\left(\frac{T^3}{n^2}\right)$ w.h.p.	if $L \leq T = O\left(n^{1/4} \log^{-1/2} n\right)$
$O(n^{3/2} \log n)$ w.h.p.	$\Omega\left(\frac{T}{n^{3/2} \log n}\right)$ w.h.p.	if $\Omega\left(n^{1/4} \log^{-1/2} n\right) = T = O\left(n^{1/4}\right)$
$O\left(\frac{n^2 \log n}{T^2}\right)$ w.h.p.	$\Omega\left(\frac{T^3}{n^2 \log n}\right)$ w.h.p.	if $\Omega\left(n^{1/4}\right) = T = O\left(n^{1/2}\right)$
$O(n \log n)$ w.h.p.	$\Omega\left(\frac{T}{n \log n}\right)$ w.h.p.	if $\Omega\left(n^{1/2}\right) = T \leq n$

5.5 Geometric Dynamic Networks

As shown in Chapter 3, geometric dynamic networks are $\left\lfloor \frac{R-1}{2v_{\max}} \right\rfloor + 1$-interval connected if $R \geq 1$ (cf. Lemma 3.1) and, if $R \geq 2$, they contain a spanning $\left\lfloor \frac{1}{2}R \right\rfloor$-vertex connected subgraph that is stable for $\left\lfloor \frac{R}{4v_{\max}} \right\rfloor + 1$ rounds (cf. Lemma 3.2). If nodes know their positions (e.g., by using Galileo, GPS or GLONASS) or if they at least have the ability to sense the distances to their neighbors, they are able to determine these stable subgraphs and the algorithms presented in this chapter can be applied. For the MIS computation, the deterministic MIS algorithm by Schneider and Wattenhofer [SW08] can be used (see Lemma 5.15 for details) since the stable subgraphs are growth-bounded. This yields improved results for geometric dynamic networks with $\mathrm{MIS}(n) = \mathrm{O}(\log^* n)$ and $T = \Theta(R \cdot v_{\max}^{-1})$.

5.6 Conclusion and Unanswered Questions

This chapter showed that extremum and summation problems can be solved faster than dissemination problems in T-stable dynamic networks. Here, properties such as commutativity, associativity, and idempotence were exploited. In particular, idempotence seems to make the extremum problem a lot simpler. Future work could focus on new problems that have different properties and allow for aggregation. It would also be interesting to see if similar techniques could be applied to other dynamic models such as T-interval connected dynamic networks. Furthermore, lower bounds for these problems are of interest. In case of the summation problem, this could lead to a nontrivial lower bound for the counting problem (cf. Unanswered Question 4.1) since the counting problem can be reduced to a summation problem, in which each node starts with a 1 as input.

CHAPTER 6

Conclusion & Outlook

This thesis delivers several insights into dynamic ad-hoc networks. First of all, the geometric dynamic network model with constant speeds is a quite modest adversary model for wireless networks, since it neglects many difficulties arising in reality. Still, the assumptions are sufficient to reproduce the $\Omega(nk)$ lower bound for knowledge-based token-forwarding algorithms, previously known for T-stable and T-interval connected dynamic networks. If speeds are smaller, then it is possible to speed up token dissemination algorithms and prove almost matching lower bounds (cf. Chapter 3).

Next, counting the number of nodes in a dynamic network—maybe the most fundamental problem one could ask to solve—is a key problem for many other problems (e.g., the noncontinuous computation of an extremum). While the exact complexity of this problem is still unknown, the construction provided in Chapter 4 could be helpful and lead to a nontrivial lower bound some day. Conversely, if the two-party k-token dissemination problem is not suited for finding a lower bound for counting, then the question of its exact complexity is intriguing as well.

Finally, the continuous computation aspects and performance metrics presented in Chapter 5 constitute a framework for the design of algorithms solving aggregation problems in sensor networks. As it turns out, the output rates of algorithms for continuous aggregation problems can be increased by applying more advanced techniques than just executing algorithms for the noncontinuous versions over and over again.

6.1 Future Research Directions

While some unanswered research questions have already been posed at the ends of Chapter 3, Chapter 4, and Chapter 5, this section discusses possible research directions that are of major interest and go beyond those questions.

The Complexity of Counting and Related Problems As stated several times, the complexity of counting plays a central role (cf. Unanswered Question 4.1). Further, it would be interesting to see if it is possible to find any nontrivial upper bound on the number of nodes in the network in $o(n^2)$ rounds. Here, nontrivial refers to a bound better than polynomial in n since a polynomial bound is revealed by the unique ID universe (as modeled in this thesis, it must have polynomial size).

Unanswered Question 6.1 (Complexity of Finding an Upper Bound on the Number of Nodes)**.** Is it possible to find a nontrivial upper bound on the number of nodes n in $o(n^2)$ rounds in 1-stable / 1-interval connected dynamic networks?

Closely related to this question is the problem of determining whether all nodes have got the same input. Note that this problem is a special extremum problem. If it can be solved in $o(n^2)$ rounds, then the number of rounds required to do this gives a nontrivial upper bound on n.

Unanswered Question 6.2 (Complexity of the Same Input Problem)**.** Is it possible to decide in $o(n^2)$ rounds whether all nodes have got the same input?

Continuous Aggregation Problems In the context of continuous aggregation, several extensions are conceivable. Firstly, the techniques presented in Chapter 5 are applicable to T-stable dynamic networks only. The question that arises is whether it is possible to apply similar ideas to T-interval connected dynamic networks. However, it is not obvious how to transfer the graph patching technique, which is a central issue for the approach used in this thesis. Secondly, further problems could be analyzed in this context. Possible candidates might be majority functions. Thirdly, only upper bounds for the output rates of several problems are known. It is not clear what the limitations of these aggregation algorithms are. Thus, it would be interesting to study the problems considered in this thesis from a lower bound perspective.

Dynamic Overlay Networks The models analyzed in this thesis are well suited for wireless networks. However, other kinds of networks are dynamic as well, even though in a very different way. Consider, for example, a network shared by different applications that use different overlay networks . From the perspective of a single overlay, the available bandwidth changes over time. This interferes with the execution of information dissemination algorithms. A worthwhile approach could be to study these overlay networks under a worst-case and omnipresent dynamic as in the dynamic network model discussed here. The main challenge will probably be to find an adequate adversary model that still allows algorithms to do something meaningful.

Bibliography

[14] "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture." In: *IEEE Std 802-2014 (Revision to IEEE Std 802-2001)* (June 2014), pp. 1–74. DOI: 10.1109/IEEESTD.2014.6847097.

[Abs+13a] S. Abshoff, M. Benter, A. Cord-Landwehr, M. Malatyali, and F. Meyer auf der Heide. "Token Dissemination in Geometric Dynamic Networks." In: *Algorithms for Sensor Systems - 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, ALGOSENSORS 2013, Sophia Antipolis, France, September 5-6, 2013, Revised Selected Papers*. Ed. by P. Flocchini, J. Gao, E. Kranakis, and F. Meyer auf der Heide. Vol. 8243. Lecture Notes in Computer Science. Springer, 2013, pp. 22–34. DOI: 10.1007/978-3-642-45346-5_3.

[Abs+13b] S. Abshoff, M. Benter, M. Malatyali, and F. Meyer auf der Heide. "On Two-Party Communication through Dynamic Networks." In: *Principles of Distributed Systems - 17th International Conference, OPODIS 2013, Nice, France, December 16-18, 2013. Proceedings*. Ed. by R. Baldoni, N. Nisse, and M. van Steen. Vol. 8304. Lecture Notes in Computer Science. Springer, 2013, pp. 11–22. DOI: 10.1007/978-3-319-03850-6_2.

[AM14] S. Abshoff and F. Meyer auf der Heide. "Continuous Aggregation in Dynamic Ad-Hoc Networks." In: *Structural Information and Communication Complexity - 21st International Colloquium, SIROCCO 2014, Takayama, Japan, July 23-25, 2014. Proceedings*. Ed. by M. M. Halldórsson. Vol. 8576. Lecture Notes in Computer Science. Springer, 2014, pp. 194–209. DOI: 10.1007/978-3-319-09620-9_16.

[Aug+12] J. Augustine, G. Pandurangan, P. Robinson, and E. Upfal. "Towards robust and efficient computation in dynamic peer-to-peer networks." In: *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*. Ed. by Y. Rabani. SIAM, 2012, pp. 551–569. DOI: 10.1137/1.9781611973099.

[Awe87] B. Awerbuch. "Optimal Distributed Algorithms for Minimum Weight Spanning Tree, Counting, Leader Election and Related Problems (Detailed Summary)." In: *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*. Ed. by A. V. Aho. ACM, 1987, pp. 230–240. DOI: 10.1145/28395.28421.

[Bie+09] M. Bienkowski, J. Byrka, M. Korzeniowski, and F. Meyer auf der Heide. "Optimal algorithms for page migration in dynamic networks." In: *J. Discrete Algorithms* 7.4 (2009), pp. 545–569. DOI: 10.1016/j.jda.2008.07.006.

[BM12] P. Brandes and F. Meyer auf der Heide. "Distributed computing in fault-prone dynamic networks." In: *4th Workshop on Theoretical Aspects of Dynamic Distributed Systems, TADDS '12, Roma, Italy, December 17, 2012*. Ed. by L. Blin and Y. Busnel. ACM, 2012, pp. 9–14. DOI: 10.1145/2414815.2414818.

[BRS12] M. Biely, P. Robinson, and U. Schmid. "Agreement in Directed Dynamic Networks." In: *Structural Information and Communication Complexity - 19th International Colloquium, SIROCCO 2012, Reykjavik, Iceland, June 30-July 2, 2012, Revised Selected Papers*. Ed. by G. Even and M. M. Halldórsson. Vol. 7355. Lecture Notes in Computer Science. Springer, 2012, pp. 73–84. DOI: 10.1007/978-3-642-31104-8_7.

[Cas+12] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. "Time-varying graphs and dynamic networks." In: *IJPEDS* 27.5 (2012), pp. 387–408. DOI: 10.1080/17445760.2012.668546.

[CD97] S. Chaudhuri and D. P. Dubhashi. "Probabilistic Recurrence Relations Revisited." In: *Theor. Comput. Sci.* 181.1 (1997), pp. 45–56. DOI: 10.1016/S0304-3975(96)00261-7.

[CGN12] A. Cornejo, S. Gilbert, and C. C. Newport. "Aggregation in dynamic networks." In: *ACM Symposium on Principles of Distributed Computing, PODC '12, Funchal, Madeira, Portugal, July 16-18, 2012*.

Ed. by D. Kowalski and A. Panconesi. ACM, 2012, pp. 195–204. DOI: 10.1145/2332432.2332468.

[Das+12] A. Das Sarma, S. Holzer, L. Kor, A. Korman, D. Nanongkai, G. Pandurangan, D. Peleg, and R. Wattenhofer. "Distributed Verification and Hardness of Distributed Approximation." In: *SIAM J. Comput.* 41.5 (2012), pp. 1235–1265. DOI: 10.1137/11085178X.

[Deg+11] B. Degener, B. Kempkes, T. Langner, F. Meyer auf der Heide, P. Pietrzyk, and R. Wattenhofer. "A tight runtime bound for synchronous gathering of autonomous robots with limited visibility." In: *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*. Ed. by R. Rajaraman and F. Meyer auf der Heide. ACM, 2011, pp. 139–148. DOI: 10.1145/1989493.1989515.

[Di +13] G. A. Di Luna, S. Bonomi, I. Chatzigiannakis, and R. Baldoni. "Counting in Anonymous Dynamic Networks: An Experimental Perspective." In: *Algorithms for Sensor Systems - 9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics, ALGOSENSORS 2013, Sophia Antipolis, France, September 5-6, 2013, Revised Selected Papers*. Ed. by P. Flocchini, J. Gao, E. Kranakis, and F. Meyer auf der Heide. Vol. 8243. Lecture Notes in Computer Science. Springer, 2013, pp. 139–154. DOI: 10.1007/978-3-642-45346-5_11.

[Di +14a] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. "Conscious and Unconscious Counting on Anonymous Dynamic Networks." In: *Distributed Computing and Networking - 15th International Conference, ICDCN 2014, Coimbatore, India, January 4-7, 2014. Proceedings*. Ed. by M. Chatterjee, J. Cao, K. Kothapalli, and S. Rajsbaum. Vol. 8314. Lecture Notes in Computer Science. Springer, 2014, pp. 257–271. DOI: 10.1007/978-3-642-45249-9_17.

[Di +14b] G. A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. "Counting in Anonymous Dynamic Networks under Worst-Case Adversary." In: *IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014, Madrid, Spain, June 30 - July 3, 2014*. IEEE Computer Society, 2014, pp. 338–347. DOI: 10.1109/ICDCS.2014.42.

[Dij74] E. W. Dijkstra. "Self-stabilizing Systems in Spite of Distributed Control." In: *Commun. ACM* 17.11 (1974), pp. 643–644. DOI: 10.1145/361179.361202.

[Dut+13] C. Dutta, G. Pandurangan, R. Rajaraman, Z. Sun, and E. Viola. "On the Complexity of Information Spreading in Dynamic Networks." In: *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*. Ed. by S. Khanna. SIAM, 2013, pp. 717–736. DOI: 10.1137/1.9781611973105.52.

[FHW12] S. Frischknecht, S. Holzer, and R. Wattenhofer. "Networks cannot compute their diameter in sublinear time." In: *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*. Ed. by Y. Rabani. SIAM, 2012, pp. 1150–1162. DOI: 10.1137/1.9781611973099.

[FRS09] H. Frey, S. Rührup, and I. Stojmenović. "Routing in Wireless Sensor Networks." In: *Guide to Wireless Sensor Networks*. Ed. by S. C. Misra, I. Woungang, and S. Misra. Computer Communications and Networks. Springer London, 2009, pp. 81–111. DOI: 10.1007/978-1-84882-218-4_4.

[HK11] B. Haeupler and D. R. Karger. "Faster information dissemination in dynamic networks via network coding." In: *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, June 6-8, 2011*. Ed. by C. Gavoille and P. Fraigniaud. ACM, 2011, pp. 381–390. DOI: 10.1145/1993806.1993885.

[HK12] B. Haeupler and F. Kuhn. "Lower Bounds on Information Dissemination in Dynamic Networks." In: *Distributed Computing - 26th International Symposium, DISC 2012, Salvador, Brazil, October 16-18, 2012. Proceedings*. Ed. by M. K. Aguilera. Vol. 7611. Lecture Notes in Computer Science. Springer, 2012, pp. 166–180. DOI: 10.1007/978-3-642-33651-5_12.

[Kar94] R. M. Karp. "Probabilistic Recurrence Relations." In: *J. ACM* 41.6 (1994), pp. 1136–1150. DOI: 10.1145/195613.195632.

[KLO10] F. Kuhn, N. A. Lynch, and R. Oshman. "Distributed computation in dynamic networks." In: *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. Ed. by L. J. Schulman. ACM, 2010, pp. 513–522. DOI: 10.1145/1806689.1806760.

[KLS08] F. Kuhn, T. Locher, and S. Schmid. "Distributed computation of the mode." In: *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008, Toronto, Canada, August 18-21, 2008*. Ed. by R. A. Bazzi and B. Patt-Shamir. ACM, 2008, pp. 15–24. DOI: 10.1145/1400751.1400756.

[KLW07] F. Kuhn, T. Locher, and R. Wattenhofer. “Tight bounds for distributed selection.” In: *SPAA 2007: Proceedings of the 19th Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Diego, California, USA, June 9-11, 2007*. Ed. by P. B. Gibbons and C. Scheideler. ACM, 2007, pp. 145–153. DOI: 10.1145/1248377.1248401.

[KM11] B. Kempkes and F. Meyer auf der Heide. “Local, Self-organizing Strategies for Robotic Formation Problems.” In: *Algorithms for Sensor Systems - 7th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, ALGOSENSORS 2011, Saarbrücken, Germany, September 8-9, 2011, Revised Selected Papers*. Ed. by T. Erlebach, S. E. Nikoletseas, and P. Orponen. Vol. 7111. Lecture Notes in Computer Science. Springer, 2011, pp. 4–12. DOI: 10.1007/978-3-642-28209-6_2.

[KMO11] F. Kuhn, Y. Moses, and R. Oshman. “Coordinated consensus in dynamic networks.” In: *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, June 6-8, 2011*. Ed. by C. Gavoille and P. Fraigniaud. ACM, 2011, pp. 1–10. DOI: 10.1145/1993806.1993808.

[KO11a] F. Kuhn and R. Oshman. “Dynamic networks: models and algorithms.” In: *SIGACT News* 42.1 (2011), pp. 82–96. DOI: 10.1145/1959045.1959064.

[KO11b] F. Kuhn and R. Oshman. “The Complexity of Data Aggregation in Directed Networks.” In: *Distributed Computing - 25th International Symposium, DISC 2011, Rome, Italy, September 20-22, 2011. Proceedings*. Ed. by D. Peleg. Vol. 6950. Lecture Notes in Computer Science. Springer, 2011, pp. 416–431. DOI: 10.1007/978-3-642-24100-0_40.

[Kuh+03] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. “Geometric ad-hoc routing: of theory and practice.” In: *Proceedings of the Twenty-Second ACM Symposium on Principles of Distributed Computing, PODC 2003, Boston, Massachusetts, USA, July 13-16, 2003*. Ed. by E. Borowsky and S. Rajsbaum. ACM, 2003, pp. 63–72. DOI: 10.1145/872035.872044.

[Kuh+10] F. Kuhn, C. Lenzen, T. Locher, and R. Oshman. “Optimal gradient clock synchronization in dynamic networks.” In: *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing, PODC 2010, Zurich, Switzerland, July 25-28, 2010*. Ed. by A. W. Richa and R. Guerraoui. ACM, 2010, pp. 430–439. DOI: 10.1145/1835698.1835799.

[KWZ03] F. Kuhn, R. Wattenhofer, and A. Zollinger. "Worst-Case optimal and average-case efficient geometric ad-hoc routing." In: *Proceedings of the 4th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2003, Annapolis, Maryland, USA, June 1-3, 2003*. ACM, 2003, pp. 267–278. DOI: 10.1145/778415.778447.

[Lin92] N. Linial. "Locality in Distributed Graph Algorithms." In: *SIAM J. Comput.* 21.1 (1992), pp. 193–201. DOI: 10.1137/0221015.

[Lub86] M. Luby. "A Simple Parallel Algorithm for the Maximal Independent Set Problem." In: *SIAM J. Comput.* 15.4 (1986), pp. 1036–1053. DOI: 10.1137/0215074.

[MCS12a] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. "Brief Announcement: Naming and Counting in Anonymous Unknown Dynamic Networks." In: *Distributed Computing - 26th International Symposium, DISC 2012, Salvador, Brazil, October 16-18, 2012. Proceedings*. Ed. by M. K. Aguilera. Vol. 7611. Lecture Notes in Computer Science. Springer, 2012, pp. 437–438. DOI: 10.1007/978-3-642-33651-5_46.

[MCS12b] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. "Causality, Influence, and Computation in Possibly Disconnected Synchronous Dynamic Networks." In: *Principles of Distributed Systems, 16th International Conference, OPODIS 2012, Rome, Italy, December 18-20, 2012. Proceedings*. Ed. by R. Baldoni, P. Flocchini, and B. Ravindran. Vol. 7702. Lecture Notes in Computer Science. Springer, 2012, pp. 269–283. DOI: 10.1007/978-3-642-35476-2_19.

[MCS13] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. "Naming and Counting in Anonymous Unknown Dynamic Networks." In: *Stabilization, Safety, and Security of Distributed Systems - 15th International Symposium, SSS 2013, Osaka, Japan, November 13-16, 2013. Proceedings*. Ed. by T. Higashino, Y. Katayama, T. Masuzawa, M. Potop-Butucaru, and M. Yamashita. Vol. 8255. Lecture Notes in Computer Science. Springer, 2013, pp. 281–295. DOI: 10.1007/978-3-319-03089-0_20.

[MCS14] O. Michail, I. Chatzigiannakis, and P. G. Spirakis. "Causality, influence, and computation in possibly disconnected synchronous dynamic networks." In: *J. Parallel Distrib. Comput.* 74.1 (2014), pp. 2016–2026. DOI: 10.1016/j.jpdc.2013.07.007.

[MS06] D. Mosk-Aoyama and D. Shah. "Computing separable functions via gossip." In: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006*. Ed. by E. Ruppert and D. Malkhi. ACM, 2006, pp. 113–122. DOI: `10.1145/1146381.1146401`.

[Osh12] R. Oshman. "Distributed Computation in Wireless and Dynamic Networks." PhD thesis. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, Sept. 2012. DOI: `1721.1/78456`.

[OW05] R. O'Dell and R. Wattenhofer. "Information dissemination in highly dynamic graphs." In: *DIALM-POMC*. 2005, pp. 104–110. DOI: `10.1145/1080810.1080828`.

[Pel00] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, 2000. DOI: `10.1137/1.9780898719772`.

[PS92] A. Panconesi and A. Srinivasan. "Improved Distributed Algorithms for Coloring and Network Decomposition Problems." In: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*. Ed. by S. R. Kosaraju, M. Fellows, A. Wigderson, and J. A. Ellis. ACM, 1992, pp. 581–592. DOI: `10.1145/129712.129769`.

[SW08] J. Schneider and R. Wattenhofer. "A log-star distributed maximal independent set algorithm for growth-bounded graphs." In: *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008, Toronto, Canada, August 18-21, 2008*. Ed. by R. A. Bazzi and B. Patt-Shamir. ACM, 2008, pp. 35–44. DOI: `10.1145/1400751.1400758`.

[Yao79] A. C. Yao. "Some Complexity Questions Related to Distributive Computing (Preliminary Report)." In: *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*. Ed. by M. J. Fischer, R. A. DeMillo, N. A. Lynch, W. A. Burkhard, and A. V. Aho. ACM, 1979, pp. 209–213. DOI: `10.1145/800135.804414`.