Defence Research and Development Canada
Recherche et développement pour la défense Canada

Deterministic algorithms in dynamic networks

Formal models and metrics

Arnaud Casteigts and Paola Flocchini

Defence R&D Canada – Ottawa

Canada

Contract Report
DRDC Ottawa CR 2013-020
April 2013

Deterministic algorithms in dynamic networks

Formal models and metrics

Arnaud Casteigts
Paola Flocchini
University of Ottawa

Prepared by:

University of Ottawa
School of Electrical Engineering and Computer Science
800 King Edward Avenue
Ottawa, Ontario
K1N 6N5

Contract Number: W7714-115111/001/SV
Contract Scientific Authority: Matthew Kellett 613-991-4362

Defence R&D Canada – Ottawa
Contract Report
DRDC Ottawa CR 2013-020
April 2013

Approved by

Original signed by Jean-François Rivest

Jean-François Rivest
Cyber Operations and Signals Warfare Section

Approved for release by

Original signed by Chris McMillan

Chris McMillan
Chief Scientist, DRDC Ottawa

Abstract

The number of telecommunication networks deployed in a dynamic environment is quickly growing. This trend exists both in everyday life (e.g., smartphones, vehicles, and commercial satellites) and in a military context (e.g., dismounted soldiers or swarms of UAVs). Unfortunately, few theoretical tools to date have enabled the study of dynamic networks in a formal and rigorous way. As a result, it is hard and sometimes impossible to guarantee, mathematically, that a given algorithm will reach its objectives once deployed in real conditions. In this report, we identify a collection of recent theoretical tools whose purpose is to model, describe, and leverage dynamic networks in a formal way. These tools include a dynamic graph formalism, various computational models, and communication models for distributed networks. We extend many graph theoretical concepts towards a dynamic variant and show how these new variants impact the solution of classical distributed problems. The report also presents a hierarchy of dynamic networks based on dynamic graph properties, thereby offering a combinatorial alternative to the well-known mobility models typically used in simulations.

Résumé

Nous assistons à une augmentation rapide du nombre de réseaux de télécommunications utilisés en environnement dynamique. Cette tendance se voit tant dans la vie civile (notamment téléphones intelligents, véhicules et satellites commerciaux) que dans le contexte militaire (comme les soldats débarqués ou les essaims d'UAV). Cependant, peu d'outils théoriques peuvent actuellement aider à étudier les réseaux dynamiques de façon formelle et rigoureuse. Il est donc difficile sinon impossible de démontrer mathématiquement qu'un algorithme particulier atteindra ses objectifs une fois déployé en situation réelle. Le présent rapport décrit plusieurs outils théoriques récemment créés servant à modéliser et décrire formellement les réseaux dynamiques et ainsi d'en mieux tirer parti. Parmi ces outils, on compte un modélisateur de graphes dynamiques, divers modèles informatiques et des modèles de communication en réseau distribué. Nous adaptons ensuite aux réseaux dynamiques plusieurs concepts en théorie des graphes, puis nous démontrons les répercussions de ces concepts adaptés sur la solution de problèmes distribués connus. Le présent rapport hiérarchise aussi les réseaux dynamiques en fonction de diverses propriétés des graphes dynamiques, et propose ainsi une solution combinatoire pouvant remplacer les modèles de mobilité connus le plus souvent utilisés dans les simulations.

This page intentionally left blank.

Executive summary

Deterministic algorithms in dynamic networks: Formal models and metrics

Arnaud Casteigts, Paola Flocchini; DRDC Ottawa CR 2013-020; Defence R&D Canada – Ottawa; April 2013.

Background: Telecommunication networks are evolving continuously and the number of these networks deployed in a dynamic environment is quickly growing. In everyday life we have networked smartphones and vehicles and in a military context there are dismounted soldiers and swarms of UAVs. Unfortunately, few theoretical tools to date have enabled the study of dynamic networks in a formal and rigorous way. The current trend in research is to rely instead on simulations, which necessarily are not fully understood, in order to assess the behaviour of a given solution in a given type of network.

Results: We identify a collection of recent theoretical tools whose purpose is to model, describe, and leverage dynamic networks in a formal way. Doing so, we reveal the potential of these tools to guarantee, mathematically, that a given algorithm will reach its objective with certainty. These theoretical tools include a dynamic graph formalism, various computational models, and communication models for distributed networks. We extend a range of graph theoretical concepts toward a dynamic variant and show how these new variants impact the solution of classical distributed problems. This report also presents a hierarchy of dynamic networks based on dynamic graph properties, thereby offering a combinatorial alternative to the well-known mobility models typically used in simulations.

Significance: Having these kinds of tools is instrumental in studying the behaviour of algorithms in dynamic networks. This fact is all the more true when the objective is to show, mathematically, that an algorithm will or will not solve a given problem in a given context. This particular concern is very appealing in a military context, where reliability can be a key attribute of military systems.

Future plans: This report focuses on models and formalisms representing networks, algorithms and related concepts. An upcoming report (Casteigts and Flocchini 2013, CR 2012-021) deals in detail with algorithmic and analytical tools that have been developed on top of these models and formalisms.

Sommaire

Deterministic algorithms in dynamic networks: Formal models and metrics

Arnaud Casteigts, Paola Flocchini ; DRDC Ottawa CR 2013-020 ; R & D pour la défense Canada – Ottawa ; avril 2013.

Contexte : Les réseaux de télécommunication évoluent continuellement, et nous assistons à une augmentation rapide du nombre de réseaux de télécommunications utilisés en environnement dynamique. Dans la vie courante, nous avons maintenant des téléphones intelligents et mme des véhicules réseautés, et, dans le contexte militaire, des soldats débarqués et des essaims d'UAV. Cependant, peu d'outils théoriques peuvent actuellement aider à étudier les réseaux dynamiques de façon formelle et rigoureuse. En recherche, on tend actuellement à utiliser des simulations, obligatoirement pas comprises parfaitement, afin d'évaluer le comportement d'une solution donnée dans un type de réseau précis.

Résultats : Le présent rapport décrit plusieurs outils théoriques récemment créés servant à modéliser et décrire formellement les réseaux dynamiques et ainsi d'en mieux tirer parti. Nous dégageons ainsi le potentiel qu'ont ces outils de démontrer mathématiquement qu'un algorithme donné va atteindre avec certitude son objectif. Parmi ces outils théoriques, on compte un modélisateur de graphiques dynamiques, divers modèles informatiques et des modèles de communication en réseau distribué. Nous adaptons ensuite aux réseaux dynamiques plusieurs concepts en théorie des graphes, puis nous démontrons les répercussions de ces concepts adaptés sur la solution de problèmes distribués connus. Le présent rapport hiérarchise aussi les réseaux dynamiques en fonction de diverses propriétés des graphes dynamiques, et propose ainsi une solution combinatoire pouvant remplacer les modèles de mobilité connus le plus souvent utilisés dans les simulations.

Importance : Disposer de ce type d'outils est vital afin de pouvoir étudier le comportement des algorithmes en réseaux dynamiques, ce qui vaut en double s'il s'agit de faire la démonstration mathématique qu'un algorithme pourra ou ne pourra pas résoudre un problème précis dans un contexte donné. Cette question se révèle particulièrement intéressante dans un contexte militaire, car la fiabilité est souvent un attribut crucial des systèmes militaires.

Perspectives : Le présent rapport est axé sur des modèles formels qui représentent des réseaux, des algorithmes et d'autres notions connexes. Un rapport subséquent (Casteigts and Flocchini 2013, CR 2012-021) traite en détail des outils algorithmiques et d'analyse qui auront été développés à l'aide de ces modèles formels.

Table of contents

List of figures

List of tables

List of algorithms

This page intentionally left blank.

A note on references

This report uses the "author-year" style of citation. As a result, when a paper involves three or more authors, only the first one appears in the citation, e.g., First *et. al.* We invite the reader to keep in mind that the tradition in theoretical computer science is to order a paper's authors alphabetically, and as such, the first author is not necessarily the "main" contributor of the cited paper. The complete list of authors of a paper can be found in the References section.

Acknowledgements

We are deeply thankful to Matthew Kellett for useful guidance and feedback on this report.

This page intentionally left blank.

1 Introduction

A revolution is taking place in communication technologies due to the increasing availability of wireless equipment and ever-shrinking computational devices. It is now possible to embed computational and communicating features in many new application contexts such as networks of vehicles, sensors, mobile robots, as well as on unmanned aerial vehicles (UAVs) or dismounted soldiers.

In general, the applications and services that are deployed in a network rely on assumptions about the network topology (e.g., What is the shape of the network? How is it organized? Is the organization stable?). In most of the new contexts, the topology is ever changing. For a majority of them, the dynamics are even unpredictable. In this seemingly chaotic environment, early algorithmic efforts have mainly been devoted to designing probabilistic solutions, whose outcome is certified with high probability. Unfortunately, military applications are often too critical to settle for the probable (or even highly probable) success of a task, when a deterministic alternative exists. Though alternatives do not always exist, when they do exist at comparable performance, they should be preferred. In some cases, one might even be willing to trade some performance for a guarantee.

In this work, we explore the basic ingredients—models, concepts, metrics—to be used in developing deterministic solutions whose correct behaviour or outcome can be mathematically proven. Finding deterministic solutions is challenging because most dynamic environments seem chaotic and unpredictable at first sight. However, even the most dynamic network is in general constrained in some way. For example, movements of the nodes might follow some trajectory (e.g., satellites), obey field constraints (e.g., road topology for vehicles), or social properties (e.g., group of people moving but remaining a group). Thus, some forms of regularity or semi-predictable properties can still be found and exploited. We are interested in understanding what these properties are.

We focus on models and metrics that are able to represent and express the kind of topological properties that could be exploited by a deterministic algorithm. It is generally admitted to describe a network topology by means of a graph whose vertices (or nodes) represent the communicating entities, and edges (or links) represent communication opportunities between these entities. Graphs are powerful abstractions, but they may hide or over-simplify important details about the real world. Therefore, in this document, we first review the main approaches to abstract the physical world. Since most, if not all, networks we consider are wireless, we focus on the ways graph topologies can be inferred from physical wireless topologies. This approach covers a variety of models, ranging from analytically-friendly but unrealistic ones like unit disk graphs (UDGs), to less trivial but more realistic models like the signal-to-interference plus noise ratio (SINR), which is garnering more and more attention lately.

After a brief overview of key results related to these physical abstraction models, we start looking at the higher level view of the problem from the perspective of graphs, while keeping in mind the implications a physical context can have on the network dynamics (e.g., in SINR, the communication itself induces an additional layer of topological dynamics). The

main purpose of our study is to identify, formulate, and correlate a set of properties that relate to the dynamics of the network topology and have the potential to be exploited by deterministic algorithms. One prerequisite is a graph formalism that is sufficiently expressive to formulate complex temporal properties and yet offer intuitive and concise notations. We describe such a formalism called time-varying graphs (TVGs).

Beyond the expression of connectivity properties on the topology, we rely on TVGs to elaborate a collection of concepts and metrics appropriate to play a role in the specification and analysis of algorithms. These include dynamic graphs concepts (e.g., temporal distance and connectivity, dynamic expansion) as well as computational models to be considered on top of such dynamic graphs. The way these elements are combined and used in an analytical setting is the subject of a subsequent report (Casteigts and Flocchini 2013, CR 2013-021); however, we provide high-level consideration to these concepts in this document.

2 Abstracting wireless networks

All of the technological contexts we envision here rely on wireless communication. Even though the wireless technology itself does vary among these contexts, e.g., *WiFi, Zigbee, Bluetooth*, they all share essential features due to their wireless nature. From an algorithmic standpoint, it seems extremely difficult (if not impossible) to deal with all the physical aspects involved in wireless communication. A large number of parameters need to be considered, including interference, obstacles, directionality, transmission power, weather conditions, or perturbation of other kinds. It is therefore necessary, and common practice, to make simplifying assumptions about the physical world so as to obtain a "combinatorial" view of the network, namely, a graph. We review below the main abstraction models considered by researchers in wireless networks. The reader is referred to the works of Díaz et al. (2011), Yick et al. (2008), and Avin et al. (2009) for further exploration of the problem.

2.1 Unit disk graph and its variants

The simplest and most widespread model is the unit disk graph (UDG), wherein two nodes can reciprocally communicate if and only if the distance between them is below a given threshold. This model received tremendous attention in early research. However, its over-simplistic features led researchers to design a number of variants, and eventually more sophisticated models like the signal-to-interference plus noise ratio (SINR) model. We review here some of the main results around UDG and its variants, as well as recent investigations on SINR.

2.1.1 Unit disk graphs

The unit disk graph model arose in the field of computational geometry, being defined as the *intersection graph* of a set of unit disks on the Euclidean plane. Graphically speaking, the original model stated that an edge exist between two points if the corresponding circles intersect. However, due to the strong analogy with wireless communication, in which the radius of the circle represent a communication range, it is now often considered that an edge exist between two given points if they lie *within* each other's surface (i.e., the circles are twice as big as the original approach). Unless otherwise mentioned, we use the second representation in this section.

The UDG model is a constraint on the more general disk graph model, in which the communication radius can be different for each node. In physical terms, the disk graph model accounts for nodes whose emission power are unequal. Both models are illustrated in Figure 1.

Due to their overwhelming simplicity, UDGs have received tremendous attention in research. In particular, their straight geometric properties enabled strong theoretical characterizations (some of which are reviewed below in Section 2.1.3). On the other hand, their excessive simplicity prevents them from capturing important aspects of the physical reality.

(a) Disk graph (b) Unit disk graph

Figure 1: *Two basic models: disk graph and unit disk graph (UDG).*

2.1.2 Variants on UDGs

Several variants on UDGs were proposed to take into account crucial aspects of physical reality that are not considered in the original model, such as uncertainty, interference, or obstacles.

Quasi-UDGs

One of the main drawbacks of the UDG model is that the probability of having a link between two nodes suddenly drops from 1 to 0 when the distance between them exceeds the threshold, while in reality the signal is fades progressively (according to the square or the cube of the distance) and with some unpredictable fluctuations at the transition zone where the signal becomes difficult to distinguish from background noise. A more realistic model in this respect is the one introduced by Barrière et al. (2001), today known as *quasi-unit disk graphs* (Kuhn et al. 2003). In quasi-UDGs, two distinct thresholds are considered, one below which communication is guaranteed, the other one above which it is impossible. In between lies a zone of uncertainty, in which the communication link may or may not exist. This model is illustrated in Figure 2.

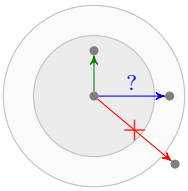


Figure 2: *Quasi-UDG.*

Interference-aware UDGs

Another important aspect of wireless communication is interference, that is, the fact that two different signals, if sent at a same time in a same area might collide and scramble each other, thereby destroying the corresponding messages. Several strategies were considered in the literature to determine which node is interferes with which other nodes. Of course, the distance between nodes is a key parameter here. The simplest solution consists in considering UDGs with the additional constraint that nodes cannot decode a message from a given neighbour if another neighbour emits at the same time.

Unfortunately, interference is likely to occur over longer distances than the communication

range represented by the unit disk. Based on this observation, a variant of UDG was proposed in which nodes can interfere within two hops (Jain et al. 2005). This model is convenient because the same graph is used for both communication and interference. However, it is not geometrically sound, as illustrated in Figure 3, based on the fact that three-hops neighbours might actually be closer than two-hops neighbours (and yet be considered as non-interfering by the model).

(a) Nodes interfere up to two hops apart.

(b) 3-hop neighbours could be closer than 2-hop neighbours.

Figure 3: *Hop-based interference. (For visibility, half-sized disks are used.)*

Another approach used for example by Alicherry et al. (2005) was to consider two distinct circles for each node (just as quasi-UDGs, but with a different meaning). Here, one of the circle corresponds to the transmission range, while the other one represents the interference range. The region between the outer circle and the inner circle represents the area where the signal is not strong enough to be received successfully, but still strong enough to interfere with other signals. Thus, two nodes are possibly interfering if they lie in the interference circle of each other. This model is illustrated in Figure 4.

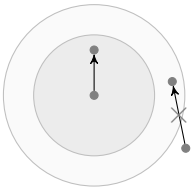


Figure 4: *Transmission radius vs. interference radius.*

Other variants on UDGs

Yet another variant of UDGs based on two different disks was used by Balasundaram and Butenko (2008). In this variant the disks represent emission radius and reception radius, respectively. Node u can communicate with node v if its emission disk intersects with v's reception disk (see Figure 5(a)).

The list of models based on UDGs could keep growing; however, most of the literature relies on those presented above. More exotic models exist for specific applications or problems, e.g., multiple directional antennas or physical obstacles, leading to a sector-based model like K-sectored UDGs (Nolan 2004), in which the space around a node is divided in a number of cones, each having distinct transmission properties (see Figure 5(b)).

(a) Emission vs. Reception radii (b) K-sectored Disk

Figure 5: *Two other variants of UDGs.*

2.1.3 Main results on UDGs

The literature around UDGs is extensive and covers several disciplines (wireless networks, computational geometry, percolation theory). Our purpose here is not to review it completely, but rather give a flavour of what type of research has been done.

Connectivity

One of the first works around unit disk graphs appears to be Gilbert (1961). Following a line of work on random graphs initiated by Erdős and Rényi (1960), he suggested that deciding what links exist in a completely random way was not realistic in most practical scenarios. In his model, points are distributed uniformly at random (*i.e.*, their positions follow a Poisson process of given density over the plane). Then two points are joined by a line if and only if they are separated by a distance less than a given threshold. The function of main interest in Gilbert's work was the probability $P(N)$ that a point belongs to a component containing at least N points. The difficulty of this type of question became instantly apparent, and although some bounds are characterized analytically, Gilbert eventually relied on simulations to approximate real threshold values (this was in 1961!).

The kind of graphs studied by Gilbert are now called *random geometric graphs*. The plane is often represented as a unit square (noted $[0,1]^2$) in which nodes are randomly distributed and have communication range lower than 1. To avoid boundary effects (i.e., the fact that points near the borders have different connectivity), it is sometimes considered to be toroidal (and thus noted $[0,1)^2$).

Most efforts in this area have focused on studying the tradeoff between communication range and connectivity. The main result, independently obtained by Penrose (1997) and by Gupta and Kumar (1998), is the existence of a connectivity threshold at $\sqrt{\frac{\ln n}{\pi n}}$, which corresponds to the expected length of the longest edge of an minimum spanning tree (a connected subset of the graph whose sum of edge lengths is minimum) on $[0,1]^2$; the threshold is asymptotically the same for $[0,1)^2$. Note that this does not guarantee the network is connected at the threshold value, but indicates the existence of a sharp transition from non-connectivity to connectivity within some additive constant in the range. Using the property that a minimal degree of k in a graph is asymptotically equivalent (i.e., it becomes equivalent as n tends to infinity) to having a k-connected graph with high probability, Penrose generalized the above threshold to the case of k-connectivity (Penrose 1999).

A graph in which the communication range is substantially higher that the connectivity threshold is called a *dense* graph. One in which the range is lower is called a *sparse* graph. The terms *superconnectivity* and *subconnectivity* can also be used. Sparse graphs in which nodes can move and "bridge" connectivity over time are of particular interest. One of the first works to study the impact mobility has on connectivity was by Grossglauser and Tse (2002).

Penrose again characterized another threshold called the *thermodynamical* threshold at $c/\sqrt{n}$ (where c is a constant) (Penrose 2003). This value corresponds to the range above which a giant component (of size $\Theta(n)$) exists with high probability. By symmetry, one can observe that below this threshold, most nodes are isolated. The constant c was experimentally determined at $c \simeq 2.0736$. More recently, Díaz et al. (2009) refined some of the above results by characterizing the probability of having a component of size i when the range is precisely at the connectivity threshold. They found the probability is $O(1/\log^i n)$, which illustrates well the sharpness of the connectivity transition.

Other problems

Many problems related to wireless networks have received tremendous attention. One may cite among others *topology control*, *transmission scheduling*, *frequency allocation*, *fault-tolerance*, *broadcasting*, *routing*, *data aggregation*, or problems related to security. Our purpose is not to review the algorithmic solutions to these problems, and we refer the interested reader to the incredible amount of literature available on the Internet on these topics. Of particular interest are problems that induced new understandings of the topological properties of wireless networks. This is the case with topology control: the problem of reducing the set of links that are used in a network so as to give this network some desired properties. The precise goal may vary with the target context or application, for example: reducing interference while maintaining connectivity (Von Rickenbach et al. 2005), assigning distinct ranges to the nodes to minimize energy consumption (Kirousis et al. 2000), or having any point of the plane covered by at least k nodes (K-coverage) (Meguerdichian et al. 2001, Wan and Yi 2006), as well as balancing communications, avoiding cycles, reducing the maximum degree in the network, etc.

For each of the problems listed above, variations exist depending on what abstraction model is considered for the network (UDG, quasi-UDG,...), which has an important impact on the solutions. Some tasks become difficult or even impossible in quasi-UDGs while being feasible in UDGs. A good example of this phenomenon is geographic routing, where guaranteed delivery requires the nodes to decide locally which subset of their neighbours must be considered as part of a planar subset of the graph (Bose et al. 1999). While feasible in UDGs, such a selection cannot be guaranteed to work in quasi-UDGs, as illustrated by the topological configuration shown in Figure 6. Here, the leftmost node discards the rightmost one as "planar" neighbour (*Gabriel* neighbour), due to the fact that two other neighbours lie within a half-circle whose diameter is the discarded edge (one for each side). However, due to the uncertainty in quasi-UDG, the right-most node might not be able to communicate with these two nodes, thereby leading to a disconnected planar subset of the graph. Based of this observation, Barrière et al. (2001) suggested an alternative approach that works in

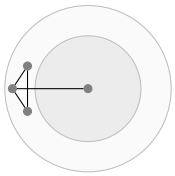


Figure 6: *Impossibility to decide planarity in quasi-UDGs (Barrière et al. 2001).*

almost all quasi-UDGs, as long as the difference between both radii does not exceed a given value (the outer radius must be no more than $\sqrt{2}$ times the inner radius).

More fundamental questions were also raised around UDGs, such as the problem of deciding whether a graph, given without geometry, can be represented as a UDG. This problem was shown to NP-hard[1] in continuous space (i.e., in case of real-valued positions) by Breu and Kirkpatrick (1998).

2.2 The SINR model

In the past few years, the need for more realistic models than UDGs and its variants became apparent. The SINR model (for signal to interference plus noise ratio) is progressively being adopted due to its more realistic features. Despite its apparent complexity, encouraging results were obtained regarding the analytical properties of this model, which seems to make it a good tradeoff between simplicity and physical realism. Surprisingly, some problems are even shown to give better solutions with SINR than with UDGs.

2.2.1 Signal to interference plus noise ratio

In the literature, SINR is often referred to as a "physical model" (as opposed to UDG, which is reffered to as a "graph-based model"). However, the SINR model's purpose is similar to UDG's in determining whether a node can send a message to another node at a given time based on the nodes' respective configurations (e.g., position, transmission power). The emphasis on the physical stems from the fact that interference and noise are directly built into the model, rather than artificially plugged in on top of it.

The SINR model can be described through a unique equation that determines whether a given node *can be heard* at a given point of the plane. Let p be that point, s_i (station i) be the node that is transmitting, and α be the fading parameter (rate at which the signal decreases with distance, usually between 2 and 3). We say that s_i is heard at p if and only if

$$\frac{E_i.dist(p,s_i)^{-\alpha}}{N+\sum_{j\neq i} E_j.dist(p,s_j)^{-\alpha}} \leq \beta \tag{1}$$

where E_k is the transmission power of node s_k, N is a background noise, and β is the reception threshold (minimal strength of signal to be heard). Observe that choosing $\beta > 1$

[1] Intuitively speaking, NP-hardness implies the execution of any solution to this problem requires an amount of time at least exponential in the number of nodes—i.e., impractical—unless the famous P = NP is proven true (something regarded as very unlikely by mathematicians and computer scientists).

directly implies that no more than one node can be heard simultaneously at a given point (i.e., the reception regions are disjoint). In that case, the reception regions partition the plane into exactly $n + 1$ surfaces: one for each emitter, plus one for the remaining surface. As explained by Avin et al. (2009), these regions converge to a Voronoi diagram if $\alpha \to \infty$ and $N = 0$ (in which case, there are eventually only n regions). If $N > 0$, each region consists of a Voronoi cell intersected with a UDG (a type of shape called "alpha shapes" in computational topology), as illustrated in Figure 7.

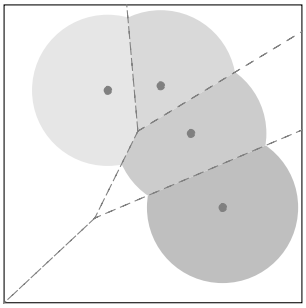


Figure 7: *Partition of the plane into $n + 1$ regions. Example by Lotker and Peleg (2010).*

On the other hand, using a value $\beta \leq 1$ makes it possible for reception regions to overlap, and thus allow several nodes to be heard at the same point. Clearly, the topology induced by SINR is of a time-dependent nature, since the potential for links depends on which nodes are communicating (assuming transmission power drops in between emissions). This characteristic makes then SINR model an interesting object of study from the perspective of communication scheduling. It also makes a terrific base for studying dynamic graph algorithms, even in the context of static networks.

2.2.2 Main results on SINR

We describe here some early results obtained with the SINR model. Most of them are mentioned in a recent survey (Lotker and Peleg 2010) to which the reader is referred for a deeper exploration. Let us start with some results regarding the comparison between UDG and SINR. Lebhar and Lotker (2009) showed that UDG could be emulated under SINR at a cost of $O(\log^3 n)$, assuming $\alpha > 2$ and the nodes are deployed uniformly at random. Informally speaking, this means that any UDG-based algorithm could also run in a SINR setting, at the expense of a "reasonable" slowdown in execution (where reasonable means here no more than a constant factor of $\log^3 n$ slower, with n being the number of nodes).

The distribution of nodes seems to play an important role in the interplay between UDG and SINR, as reinforced by the result of Moscibroda (2007), who showed that the capacity of a SINR network could be exponentially higher (in the number of nodes) than that of the corresponding UDG if specific distributions of nodes are considered. This capacity is however of the same order under a uniform distribution (Gupta and Kumar 2000).

One of the most studied problem in SINR is *communication scheduling*: the problem of planning when packets should be sent in the networks, based on a given set of communication

requests. The fact that the network topology varies with communications makes this problem all the more interesting. Scheduling in general was shown to be NP-hard both in UDG and SINR. Even *single step* scheduling—assigning the best transmission powers to a set of transmitters so as to maximize the number of simultaneous successful emitter/receiver pairs—is NP-hard, although it approximates well (Andrews and Dinitz 2009). A specific variant of *topology control* also makes particular sense in SINR, namely, that of determining how and how fast a given topology could be realized on top of a given SINR network (Moscibroda et al. 2006).

As with the UDG model a decade ago (see Section 2.1), researchers are now finding more and more structural properties of the SINR model that have algorithmic implications. The above results give some examples, another one is the concept of *fatness* of a SINR region, introduced by Avin et al. (2009) and defined as the ratio between the radius of the enclosing circle (smallest circle containing the whole region) and that of the enclosed circle (largest circle included in the region). A region is said to be fat if $\frac{enclosing\ radius}{enclosed\ radius} = O(1)$, that is, the ratio is contained within a given constant. They show that having regions that are both convex and fat is a nice property that can boost some topology-related computations, such as deciding whether a given station is heard at a given point (i.e., point location queries). Avin et al. (2009) show that if $\alpha = 2$, $\beta \geq 1$, and the network is uniform (i.e., all the nodes have identical communication power) then every region is convex and fat.

Research in SINR is still in its infancy; however, these results show the domain is theoretically fertile despite being physically realistic, which is often not the case in theoretical computer science. Some of these results even imply that the SINR model allows better solutions than UDG-based models.

3 Dynamic graphs models

A vast majority of real-world networks are dynamic, and this is especially true for wireless networks. In a majority of these networks, the dynamics are due to the movements of the nodes, which move in and out of communications range of each other as their position evolves over time. However, as observed above (see Paragraph 2.2.1), even *static* wireless networks should be considered as dynamic under realistic abstraction models like SINR. Sensor networks are another example of static networks in which the resulting topology may be dynamic due to duty cycles (energy-saving sleeping schedule) or failure.

In this section, we review some recent developments on dynamic graph theory. We first give a thorough literature review on the different attempts and motivations to model dynamic networks, which emerged in various fields of research (Section 3.1). We then describe in Section 3.2 a recently introduced framework, called time-varying graphs (TVGs), whose purpose is to integrate the existing models, concepts, and results proposed in the literature so far (Casteigts et al. 2012c). Using it, it is possible to express directly in the same formalism not only the concepts common to all these different areas, but also those specific to each. In particular, TVGs allow the identification and formal expression of those concepts and properties related to the dynamics of the graph, which can play an important role on the algorithmic side. The TVG formalism has already been used in a variety of contexts, e.g. Greve et al. (2011), Floriano et al. (2011), Zhang et al. (2012).

The formalism is then used and extended in Section 3.3, where we present the most central concepts and metrics that have been identified by the research so far (*e.g.*, journey, temporal subgraphs, distance and connectivity); we also address the different perspectives from which a graph can be looked at, such as the graph-centric (or global) point of view or the edge-centric (or interaction based) point of view.

3.1 Literature review on dynamic graphs

In the past few years, intensive research efforts have been devoted to some apparently unrelated areas of dynamic systems, obtaining closely related insights. This is particularly true for (a) the study of communication in wireless mobile ad hoc networks, e.g., broadcasting and routing in delay-tolerant networks (DTNs); (b) the exploitation of passive mobility, e.g., the opportunistic use of transportation networks; and (c) the analysis of complex real-world networks ranging from neuroscience or biology to transportation systems or social studies, e.g., the characterization of the interaction patterns emerging in a social network.

As part of these efforts, many important concepts have been identified, and sometimes formally defined. It is becoming apparent that these concepts are strongly related. In fact, in several cases, differently named concepts identified by different researchers are actually one and the same concept. For example, the concept of *temporal distance*, formalized by Bui-Xuan et al. (2003), is the same as *reachability time* (Holme 2005), *information latency* (Kossinets et al. 2008), and *temporal proximity* (Kostakos 2009). Similarly, the concept of a *journey* (Bui-Xuan et al. 2003) is called a *schedule-conforming path* (Berman 1996), a

time-respecting path (Holme 2005, Kempe et al. 2000), and a *temporal path* (Chaintreau et al. 2008, Tang et al. 2010b). Hence, the notions discovered in these investigations can be viewed as parts of the same conceptual universe, and the formalisms proposed so far to express some specific concepts can be viewed as fragments of a larger formal description of this universe. A common trait in all these areas is that the system structure—*the network topology*—varies in time. Furthermore, the rate and/or degree of the changes is considered too high to be reasonably modeled in terms of network faults: in these systems *changes are not anomalies but rather are a integral part of the nature of the system.*

As the notion of (static) graph is the natural means for representing a static network, the notion of dynamic (or *time-varying*, or *evolving*) graph is the natural means to represents these highly dynamic networks. All the concepts and definitions advanced so far imply such a notion, e.g., Kempe et al. (2000) talk of a *temporal network* (G, λ) where λ is a *time-labeling* of the edges, that associates *punctual* dates to represent dated interactions; Leskovec et al. (2007) talk of *graphs over time*; Ferreira (2004) views the dynamic of the system in terms of a sequence of static graphs, called an *evolving graph*; Flocchini et al. (2009b) employ the term *time-varying graphs*; (Kostakos 2009) uses the term *temporal graph*; etc.

The need for dynamics-related concepts emerged from a range of very different works. We mention below three research areas in which dynamical aspects have played a central role recently. They include *delay-tolerant networks*, *opportunistic-mobility networks*, and *real-world complex networks*. Interestingly, these areas have seen a number of similar concepts emerge with distinct purposes, ranging from the design of solutions in delay-tolerant networks to the analysis of phenomena in complex dynamic networks.

3.1.1 Delay-tolerant networks

Delay-tolerant networks (DTNs) are highly-dynamic, infrastructure-less networks whose essential characteristic is a possible absence of an end-to-end communication route at any instant. These networks, also called *disruption-tolerant*, *challenged*, or *opportunistic*, include for instance satellite, pedestrian, and vehicular networks. Although the assumption of *connectivity* does not necessarily hold at a given instant—the network could even be disconnected at every time instant—communication routes are generally available over time and space, enabling for example broadcast and routing by means of a store-carry-forward-like mechanism.

An extensive amount of research has been recently devoted to these types of problems. See among others the work of Burgess et al. (2006), Cardei et al. (2007), Jacquet et al. (2010), Jain et al. (2004), Liu and Wu (2009), Mahéo et al. (2008), Ruiz et al. (2012), Spyropoulos et al. (2005), Zhang (2006). A number of new routing and broadcast techniques were designed to face such an extreme situation, based for example on pro-active knowledge on the network schedule (Jain et al. 2004, Bui-Xuan et al. 2003), probabilistic strategies (Lindgren et al. 2003, Spyropoulos et al. 2005), delay-based optimization (Ros et al. 2012), or encounter-based decisions (Grossglauser and Vetterli 2003, Jones et al. 2007). Other recent works considered the broadcast problem from an analytical and probabilistic standpoint. For example Clementi

et al. (2008) and Baumann et al. (2009) characterized the maximal propagation speed as a function of the rate of topological changes in the network (these changes are themselves regulated by Markovian processes on edges). In all these investigations, the time dimension has had a strong impact on the research, and led the research community to extend most usual graph concepts—e.g., paths and reachability (Berman 1996, Kempe et al. 2000), distance (Bui-Xuan et al. 2003), diameter (Chaintreau et al. 2008), or connected components (Bhadra and Ferreira 2003)—to a temporal version.

3.1.2 Opportunistic-mobility networks

As mobile carriers and devices become increasingly equipped with short-range radio capabilities, it is possible to exploit the (delay-tolerant) networks created by their mobility for uses that are possibly external and extraneous to the carriers. In fact, other entities (e.g., code, information, web pages) called *agents* can opportunistically "move" on the carriers' network for their own purposes, by using the mobility of the carriers (sometimes called *ferries*) as a transport mechanism. Such networks have been deployed e.g., in the context of buses (Balasubramanian et al. 2007, Burgess et al. 2006) and pedestrians (Chaintreau et al. 2007). Examples of carrier networks and opportunistic mobility usages include: *Cabernet*, deployed in 10 taxis running in the Boston area (Eriksson et al. 2008), which allows the delivery of messages and files to users in cars; and *UMass DieselNet*, consisting of WiFi nodes attached to 40 buses in Amherst, used for routing, information delivery, and connectivity measurements (Burgess et al. 2006, Zhang et al. 2007).

Of particular interest is the class of carriers/ferries following a deterministic periodic trajectory. This class naturally includes infrastructure-less networks where mobile entities have fixed routes that they traverse regularly. Examples of such common settings are public transports, low earth orbiting satellite systems, security guards' tours, etc. These networks have been investigated with respect to routing and to the design of carriers' routes (Guo and Keshav 2007, Liu and Wu 2009) and more specifically for buses (Balasubramanian et al. 2007, Zhang et al. 2007) and satellites (Wang et al. 2009). In addition to routing, more analytical works around algorithmic problems have been done in the contexts of network exploration (Flocchini et al. 2009b, Ilcinkas and Wade 2011, Flocchini et al. 2009a, 2012, Kellett 2012) and the creation of broadcast structures (Casteigts et al. 2010, 2011, 2012b). In the derivation of these results, the temporal component has played a crucial role, both in terms of extending concepts and of developing new solution techniques.

3.1.3 Real-world complex networks

The research area of *complex systems* addresses the analysis of real complex dynamic networks, ranging from neuroscience and biology to transportation networks and social studies, with a particular interest in the understanding of self-organisation, emergence properties, and their reification.

As stated by Leskovec et al. (2010), the central problem in this area is the definition of mathematical models able to capture and to reproduce properties observed on the real dynamics of the networks, e.g., shrinking diameter (Leskovec et al. 2007), the formation of

communities (Börner et al. 2007, Blondel et al. 2008, Alvarez-Hamelin et al. 2008, Friggeri et al. 2011), and the appearance of inequalities. A fundamental work on graphs where edges are endowed with temporal properties is the one by Kempe and Kleinberg (2002), in which the basic properties (both combinatorial and algorithmic) of graphs are addressed when the connections among nodes are constrained by temporal conditions. The formalism introduced therein to represent dynamic graphs has been used as framework for several other works in complex systems (Backstrom et al. 2006, Eagle and Sandy, Kempe et al. 2003, Scherrer et al. 2008).

Kostakos (2009) proposes a theoretical framework for *temporal graphs* to study a large dataset of emails records. The author suggests to label graphs with temporal attributes by allowing the representation of each node as a chain of all its temporal instances during time; some interesting metrics aimed at capturing the interactions among nodes during time, e.g., temporal or geodesic proximity, are discussed. Tang et al. (2010b) propose an extension of the model of Kempe and Kleinberg (2002) by looking at the smallest delay path in a generic information spreading process. The authors try to overcome the limits of the previous works (mainly concerned with local aspects) by defining a temporal graph as a sequence of static graphs whose elements aggregate all interactions during given time-windows—we call such construct a *sequence of footprints*. Kossinets et al. (2008) study the temporal dynamics of communication over a dataset of on-line communications and emails over a two years period. The main metric introduced to capture the interaction is again the temporal distance, defined there as the minimum time needed for a piece of information to spread from an individual to another by means of multihop sequences of emails.

As these investigations indicate, temporal concerns are an integral part of recent research efforts in complex systems. It is also apparent that the emerging concepts are in essence the same as those from the field of communication networks, involving again temporal definitions of the notions of paths, distance, and connectivity, as well as higher concepts that we identify further below.

3.2 The TVG formalism

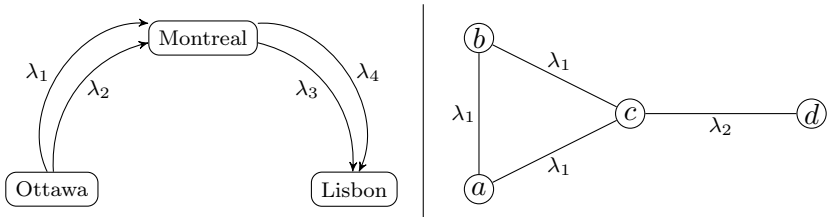
Consider a set of entities V (or *nodes*), a set of relations E between these entities (*edges*), and an alphabet L, which could describe any property such a relation could have (*label*); that is, $E \subseteq V \times V \times L$. The definition of L depends on the domain, and therefore is left open—a label could represent for example a type of carrier in a transportation network, the intensity of relation in a social network, or a medium in a communication network; in some contexts, L could be empty (and thus possibly omitted). In general, we assume L to possibly contain multi-valued elements (e.g., $\langle$*bandwidth of 4 MHz; satellite link; encryption available;...*$\rangle$). The set E enables multiple relations between a pair of entities, as long as these relations have a distinct label.

Since we address dynamical systems, the relations between entities are assumed to take place over a time span $\mathcal{T} \subseteq \mathbb{T}$ called the *lifetime* of the system. The temporal domain $\mathbb{T}$ is generally assumed to be $\mathbb{N}$ for discrete-time systems or $\mathbb{R}^+$ for continuous-time systems. The dynamics of the system can be described by a TVG, $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, where

- $\rho : E \times \mathcal{T} \to \{0, 1\}$, called *presence* function, indicates whether a given edge is available at a given time.
- $\zeta : E \times \mathcal{T} \to \mathbb{T}$, called *latency* function, indicates the time it takes to cross a given edge if starting at a given date (the latency of an edge could vary in time).

The model can be naturally extended by adding a *node presence function* $\psi : V \times \mathcal{T} \to \{0, 1\}$ (i.e., the presence of a node is conditional upon time) and a *node latency function* $\varphi : V \times \mathcal{T} \to \mathbb{T}$ (accounting e.g., for local processing times).

The TVG formalism can arguably describe a multitude of different scenarios, from transportation networks to communication networks, complex systems to social networks. Two intuitive examples are shown in Figure 8.



(a) Transportation network (b) Communication network

Figure 8: *Two examples of TVGs, employed in different contexts. Picture from (Casteigts et al. 2012c).*

The meaning of what is an edge in these two examples varies drastically. In Figure 8(a), an edge from a node u to another node v represents the possibility for some agent to *move* from u to v. The edges in this example are assumed to be directed, and possibly multiple. The meaning of the labels λ_1 to λ_4 could be, for instance, "*bus*", "*car*", "*plane*", "*boat*", respectively. Except for the travel in car from Ottawa to Montreal—which could be started anytime—typical edges in this scenario are available on a *punctual* basis, i.e., the presence function ρ for these edges returns 1 only on a particular date or dates when the trip can be started. The latency function ζ may also vary from one edge to another, as well as for different availability dates of a same given edge (e.g., variable traffic on the road, depending on the departure time).

The second example in Figure 8(b) represents a history of connectivity between a set of moving nodes, where the possibilities of communication appear, e.g., as a function of their respective distance. The two labels λ_1 and λ_2 may represent different types of communication media, such as satellite and WiFi, having various properties in terms of range, bandwidth, latency, or energy consumption. In this setting, the edges are assumed to be undirected and there cannot be more than one edge between any two nodes. The meaning of an edge is different from the previous scenario: an edge between two nodes means that either of them (or both) can attempt to send a message to the other. A typical presence function for this type of edge returns 1 for some *intervals* of time, because the nodes are typically in range for a continuous interval of time. Note that the effective delivery of a message sent at time t

on an edge e could be subjected to further constraints regarding the latency function, such as the condition that $\rho(e)$ returns 1 for the whole interval $[t, t + \zeta(e, t))$.

These two examples illustrate the spectrum of *models* over which the TVG formalism can stretch. As mentioned, some contexts are intrinsically simpler than others and require restrictions (e.g., between any two nodes in the second example, there is at most one undirected edge). Further restrictions may be considered. For example, the latency function could be constant over time ($\zeta : E \to \mathbb{T}$); over the edges ($\zeta : \mathcal{T} \to \mathbb{T}$); over both ($\zeta \in \mathbb{T}$); or simply ignored. In the last case, a TVG could have its relations fully described by a graphical representation like that of Figure 9.

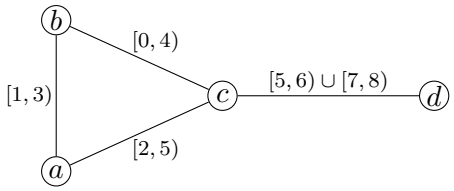


Figure 9: *A simple TVG. The interval or intervals on each edge e represent the periods of time when the edge is available, that is, $\cup(t \in \mathcal{T} : \rho(e, t) = 1)$. Example from (Casteigts et al. 2012b).*

Several analytical works on dynamic networks simply ignore ζ, or assume a discrete-time scenario where every time step implicitly corresponds to a constant ζ. This value is also usually neglected when the graph represents dated interactions over a social network (the edges in this context are generally assumed to be punctual both in terms of instantaneous presence and null latency). The definitions we give in this document address the general case, where $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$.

3.3 Temporal concepts and metrics

This section transposes and generalizes a number of dynamic network concepts into the framework of TVGs. A majority of these concepts emerged independently in various areas of the scientific literature; some appeared more specifically in the fields of telecommunication networks or distributed computing.

3.3.1 The underlying graph G

Given a TVG $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, the graph $G = (V, E)$ is called *underlying* graph of $\mathcal{G}$. This static graph should be seen as a sort of *footprint* of $\mathcal{G}$, which flattens the time dimension and indicates only the pairs of nodes that have a relation (an edge between them) at some time in $\mathcal{T}$. This is a central concept that is heavily used in the following.

In most studies and applications, G is assumed to be connected, but this is not imposed by the formalism itself. Let us stress that the connectivity of $G = (V, E)$ does not imply that $\mathcal{G}$ is connected at a given time instant; in fact, $\mathcal{G}$ could even be disconnected at all times. The lack of relationship, with regards to connectivity, between $\mathcal{G}$ and its footprint G is even

stronger: the fact that $G = (V, E)$ is connected does not even imply that $\mathcal{G}$ is "connected over time", as illustrated n Figure 10.

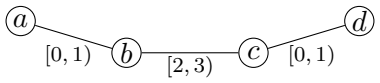


Figure 10: *An example of TVG that is not "connected over time", although its underlying graph G is connected. Here, the nodes a and d have no mean to reach each other through a chain of interaction. Picture from (Casteigts et al. 2012c).*

3.3.2 Point of views

Depending on the problem under consideration, it may be convenient to look at the evolution of the system from the point of view of a given relation (edge), or of a given entity (node), or from that of the global system (entire graph). We respectively qualify these views as *edge-centric*, *vertex-centric*, and *graph-centric*.

Edge-centric evolution

From an edge point of view, the notion of evolution comes down to a variation of availability and latency over time. We define the *available dates* of an edge e, denoted as $\mathcal{I}(e)$, as the union of all dates when the edge is available, that is, $\mathcal{I}(e) = \{t \in \mathcal{T} : \rho(e, t) = 1\}$. When $\mathcal{I}(e)$ is expressed as a multi-interval of availability $\mathcal{I}(e) = \{[t_1, t_2) \cup [t_3, t_4)...\}$, where $t_i < t_{i+1}$, we refer to the odd numbered dates $t_1, t_3, ...$ as the *appearance dates* of e and denote the sequence as $App(e)$, and we refer to the even numbered dates $t_2, t_4, ...$ as the *disappearance dates* of e and denote the sequence as $Dis(e)$. Finally, we refer to the dates themselves $t_1, t_2, t_3, ...$ as the *characteristic dates* of e, and denote the sequence as $\mathcal{S}_{\mathcal{T}}(e)$. In the following, we use the notation $\rho_{[t,t')}(e) = 1$ to indicate that $\forall t'' \in [t, t'), \rho(e, t'') = 1$.

Vertex-centric evolution

From a node standpoint, the evolution of the network consists of a succession of changes among its neighbourhood. This point of view does not appear frequently in the literature. It was used, for example, by O'Dell and Wattenhofer (2005) to express dynamic properties in terms of local variations on the *sequence of neighbourhoods* $N_{t_1}(v), N_{t_2}(v)..$ where $N_t(v)$ denotes the neighbours of v at time t and each t_i corresponds to a date of local change (i.e., appearance/disappearance of an incident edge).

The *degree* of a node u can be defined both in punctual or integral terms, e.g., with $Deg_t(u) = |E_t(u)|$, or $Deg_{\mathcal{T}}(u) = |\cup\{E_t(u) : t \in \mathcal{T}\}|$ where $E_t(u)$ indicates the set of edges incident on u at time t.

Graph-centric evolution

The sequence $\mathcal{S}_{\mathcal{T}}(\mathcal{G}) = \text{sort}(\cup\{\mathcal{S}_{\mathcal{T}}(e) : e \in E\})$, called *characteristic dates* of $\mathcal{G}$, corresponds to the sequence of dates when topological events (appearance/disappearance of an edge)

occur in the system. Each topological event can be viewed as the transformation from one static graph to another. Hence, the evolution of the system can be described as a sequence of static graphs. More precisely, from a global point of view, the evolution of $\mathcal{G}$ can be described as the sequence of graphs $\mathcal{S}_{\mathcal{G}} = G_1, G_2, \ldots$, where G_i corresponds to the static *snapshot* of $\mathcal{G}$ at time $t_i \in \mathcal{S}_{\mathcal{T}}(\mathcal{G})$; i.e., $e \in E_{G_i} \iff \rho_{[t_i,t_{i+1})}(e) = 1$. Note that, by definition, $G_i \neq G_{i+1}$.

In the case where time is discrete, another possible global representation of the evolution of $\mathcal{G}$ is the sequence $\mathcal{S}_{\mathcal{G}} = G_1, G_2, \ldots$, where G_i corresponds to the static *snapshot* of $\mathcal{G}$ at time $t = i$. In this case, it is possible that $G_i = G_{i+1}$.

Observe that in both continuous and discrete cases, the underlying graph G (defined in Section 3.3.1) corresponds to the union of all G_i in $\mathcal{S}_{\mathcal{G}}$.

The idea of representing a dynamic graph as a sequence of static graphs, mentioned in the conclusion of a paper by Harary and Gupta (1997), was brought to life by Ferreira (2004) as a combinatorial model called *evolving graphs*. An evolving graph usually refers to either one of the two structures $(G, \mathcal{S}_{\mathcal{G}}, \mathcal{S}_{\mathcal{T}})$ or $(G, \mathcal{S}_{\mathcal{G}}, \mathbb{N})$, the latter used only when discrete-time is considered. Their initial version also included a latency function, which makes them a valid—graph-centric—representation of TVGs.

3.3.3 Subgraphs of a time-varying graph

Subgraphs of a TVG $\mathcal{G}$ can be defined in a classical manner, by restricting the set of vertices or edges of $\mathcal{G}$. More interesting is the possibility to define a *temporal subgraph* by restricting the lifetime $\mathcal{T}$ of $\mathcal{G}$, leading to the graph $\mathcal{G}' = (V, E', \mathcal{T}', \rho', \zeta')$ such that

- $\mathcal{T}' \subseteq \mathcal{T}$
- $E' = \{e \in E : \exists t \in \mathcal{T}' : \rho(e,t) = 1 \wedge t + \zeta(e,t) \in \mathcal{T}'\}$
- $\rho' : E' \times \mathcal{T}' \to \{0,1\}$ where $\rho'(e,t) = \rho(e,t)$
- $\zeta' : E' \times \mathcal{T}' \to \mathbb{T}$ where $\zeta'(e,t) = \zeta(e,t)$

In practice, we allow $\mathcal{G}' = \mathcal{G}_{[t_a,t_b)}$ to denote the temporal subgraph of $\mathcal{G}$ restricted to $\mathcal{T}' = \mathcal{T} \cap [t_a, t_b)$, which includes the possible notations $\mathcal{G}_{[t_a,+\infty)}$ or $\mathcal{G}_{(-\infty,t_b)}$ to denote the temporal subgraphs of $\mathcal{G}$ going from t_a to the end of its lifetime, or from the beginning of its lifetime to t_b, regardless of whether $\mathcal{T}$ is open, semi-closed, or closed.

3.3.4 Journeys

A sequence of ordered pairs $\mathcal{J} = \{(e_1,t_1), (e_2,t_2) \ldots, (e_k,t_k)\}$, such that $\{e_1, e_2, ..., e_k\}$ is a walk in G is a *journey* in $\mathcal{G}$ if and only if $\rho(e_i,t_i) = 1$ and $t_{i+1} \geq t_i + \zeta(e_i,t_i)$ for all $i < k$. Additional constraints may be required in specific domains of application, such as the condition $\rho_{[t_i,t_i+\zeta(e_i,t_i))}(e_i) = 1$ in communication networks (the edge remains present until the message is delivered).

We denote by $departure(\mathcal{J})$, and $arrival(\mathcal{J})$, the starting date t_1 and the last date $t_k + \zeta(e_k, t_k)$ of a journey $\mathcal{J}$, respectively. Journeys can be described as *paths over time* from a source to a destination and therefore they have a *topological* length as well as a *temporal* length. The *topological length* of $\mathcal{J}$ is the number $|\mathcal{J}| = k$ of ordered pairs in $\mathcal{J}$ (i.e., the number of *hops*); its *temporal length* is its end-to-end duration: $arrival(\mathcal{J}) - departure(\mathcal{J})$.

Let us denote by $\mathcal{J}^*_{\mathcal{G}}$ the set of all possible journeys in a TVG $\mathcal{G}$, and by $\mathcal{J}^*_{(u,v)} \subseteq \mathcal{J}^*_{\mathcal{G}}$ the journeys that start at node u and terminate at node v. If a journey exists from u to v, that is, if $\mathcal{J}^*_{(u,v)} \neq \emptyset$, then we say that u can *reach* v, and allow the simplified notation $u \rightsquigarrow v$. Clearly, the existence of journey is not a symmetrical concept: $u \rightsquigarrow v \not\Leftrightarrow v \rightsquigarrow u$; this holds regardless of whether the edges are directed or not, because the time dimension creates its own level of direction. Given a node u, the set $\{v \in V : u \rightsquigarrow v\}$ is called the *horizon* of u.

We say that a journey is *direct* if the presence of consecutive edges of the journey overlap in time and the use of the subsequent edge follows directly the use of the previous edge (i.e., intermediate nodes do not wait to forward the message); otherwise, the journey is *indirect* (i.e., at least one intermediate node needs to buffer the message for some time). An example of a direct journey is $\mathcal{J}_1 = \{(ab, 2), (bc, 2 + \zeta)\}$ in the graph in Figure 9 . Examples of indirect journeys include $\mathcal{J}_2 = \{(ac, 2), (cd, 5)\}$, and $\mathcal{J}_3 = \{(ab, 2), (bc, 2 + \zeta), (cd, 5)\}$ in the same graph.

The distinction between direct and indirect journeys was suggested by Casteigts et al. (2011) to facilitate the computation of temporal distances between nodes (described next). Its implications have also been studied from the perspective of environment complexity, and more precisely to quantify how the power of an adversary controlling the links is impacted by the ability for nodes to buffer a message or not (Casteigts et al. 2012a). The set of languages that can be generated by direct journeys (no waiting allowed) contains all computable languages, whereas the set of language if waiting is allowed (indirect journeys are possible) it is just the family of *regular* languages. In other words, when waiting is not forbidden, the expressivity of the environment drops drastically from being as powerful as a Turing machine, to becoming that of a Finite-State machine, which gives an idea of the importance of buffering in dynamic networks.

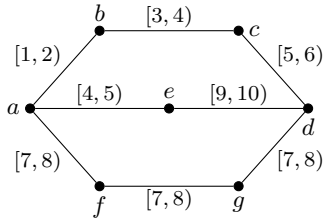
3.3.5 Distance and related metrics

As observed, the length of a journey can be measured both in terms of hops or time. This gives rise to two distinct definitions of *distance* in a TVG $\mathcal{G}$:

- The *topological distance* from a node u to a node v at time t, denoted by $d_{u,t}(v)$, is defined as $Min\{|\mathcal{J}| : \mathcal{J} \in \mathcal{J}^*_{(u,v)}, departure(\mathcal{J}) \geq t\}$. For a given date t, a journey whose departure is $t' \geq t$ and whose topological length is equal to $d_{u,t}(v)$ is qualified as *shortest*.
- The *temporal distance* from u to v at time t, denoted by $\hat{d}_{u,t}(v)$ is defined as $Min\{arrival(\mathcal{J}) : \mathcal{J} \in \mathcal{J}^*_{(u,v)}, departure(\mathcal{J}) \geq t\} - t$. Given a date t, a journey whose departure is $t' \geq t$ and arrival is $t + \hat{d}_{u,t}(v)$ is qualified as *foremost*. Finally,

for any given date t, a journey whose departure is $\geq t$ and temporal length is $Min\{\hat{d}_{u,t'}(v) : t' \in \mathcal{T} \cap [t, +\infty)\}$ is qualified as *fastest*.

These distance metrics are illustrated in Figure 11.



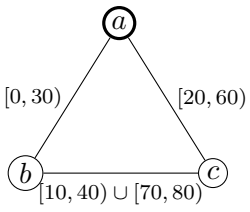
Optimal journeys between a and d (starting at time 0):

- the shortest: a-e-d (only two hops)
- the foremost: a-b-c-d (arriving at $5 + \epsilon$)
- the fastest: a-f-g-d (no intermediate waiting)

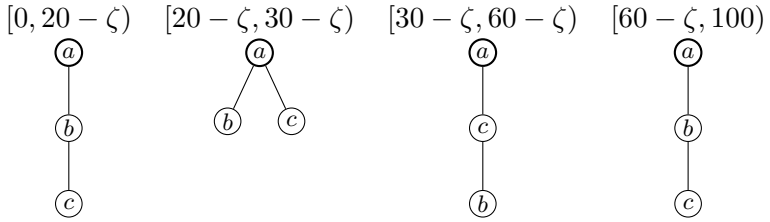
Figure 11: *Different meanings for length and distance.*

The problem of computing shortest, fastest, and foremost journeys in DTNs was introduced by Bui-Xuan et al. (2003), and an algorithm for each of the three metrics was provided for the *centralized* (combinatorial) version of the problem, assuming complete knowledge of $\mathcal{G}$. The distributed variant of this problem, namely shortest, fastest, and foremost broadcast (with termination detection at the emitter) was addressed through a line of works by Casteigts et al. (2010, 2011, 2012b).

It is important to keep in mind that the solutions to some of these problems are *time-dependent*; that is, their optimality depends on when the broadcast is initiated. An example is given in Figure 12, showing a set of solutions to the foremost broadcast problem in a simple periodically-varying graph.



(a) A simple periodic graph with period 100.



(b) Foremost broadcast trees for all possible emission dates, considering node a as the emitter. (The solution holds modulo 100.)

Figure 12: *Example of foremost broadcast trees, which are time-dependent entities. Picture from (Casteigts et al. 2012b).*

A concept closely related to the one of temporal distance is that of *temporal view*, introduced in a context of social network analysis (Kossinets et al. 2008). The temporal view[2] that a node v has of another node u at time t, denoted by $\phi_{v,t}(u)$, is defined as the latest (i.e., largest) $t' \leq t$ when a message received by v at time t could have been emitted at u; that is, in our formalism

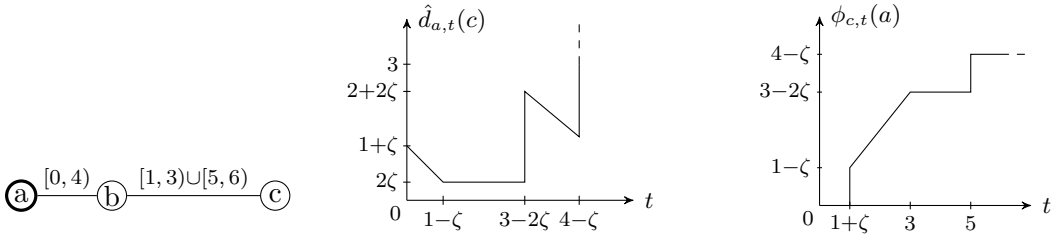
$$\phi_{v,t}(u) = \text{Max}\{departure(\mathcal{J}) : \mathcal{J} \in \mathcal{J}^*_{(u,v)}, arrival(\mathcal{J}) \leq t\}.$$

[2]This concept was called "view", but since the term has a very different meaning in distributed computing (e.g., (Yamashita and Kameda 1996)) the adjective "temporal" has been added to avoid confusion.

This definition can be seen as a generalization of the one from Kossinets et al. (2008) where only punctual contacts were considered (i.e., contacts without duration). There is a clear connection between temporal distances and temporal views. In fact both refer to the same quantity seen from different perspectives: the temporal distance is a *duration* defined locally at an *emitter* for a specific emission date, while the temporal view is a *date* defined locally at a *receiver* for a specific reception date. In fact, we have

$$\hat{d}_{u,t_e}(v) = t_r - \phi_{v,t_r}(u) \tag{2}$$

where t_e is an emission date, and t_r is the corresponding earliest reception date. As pointed out in a recent paper (Casteigts et al. 2010), these notions are deeply impacted by the co-existence of direct and indirect journeys. Indeed, the existence of arbitrarily long contacts between nodes makes it possible for adjacent edges to overlap in time and thus produce more complex patterns of time lags between nodes. Consider the plots in Figure 13, showing an example of evolution of temporal distance (from a to c) and the corresponding temporal view (that c has of a) in a very simple TVG. Contrary to the case with punctual contacts—where evolution occurs only in discrete steps—there is here a mixture of discrete and continuous evolution.



(a) A simple TVG (b) Temporal distance from a to c (c) Temporal view that c has of a

Figure 13: *Temporal distance and temporal views as a function of time (with $\zeta \ll 1$). Example from (Casteigts et al. 2012b).*

The problem of measuring temporal views in a social network was considered in the case of *punctual* contacts by Kossinets et al. (2008) based on the post-processing of email datasets. The *distributed* version of this problem in the more general case of arbitrarily long contacts (case depicted in Figure 13) was recently studied by Casteigts et al. (2011). The authors asked whether each node in a network could track in real-time how "out-of-date" it is with respect to every other node. Although relatively straight forward when punctual contacts are considered, this problem becomes substantially more complex in the case of arbitrarily long contacts due to the heterogeneous evolution of the views (continuous, based on direct journeys, and discrete, based on indirect journeys). The problem was further complicated (but also made more realistic) by considering *continuous-time* systems and *non-nil* message latency ζ; however, this latency was assumed to be constant for all links and known to the nodes. It was demonstrated that the problem remains solvable in this context by generalizing a time-measurement vector clock construct to the case of "non-punctual causality", resulting in a tool called T-Clocks.

The purpose of T-Clocks (of which one instance must run in each node) is to track two kinds of variables: the *level* of the underlying node with respect to any other node (that is, the topological length of the shortest *direct* journey currently arriving, if any, from any remote node), and the largest *date* at which a message carried to the local node through an *indirect* journey could have been emitted at any other node. T-Clocks can then serve as an abstraction layer between the network and some higher algorithm (see Figure 14), which uses temporal view information to solve more concrete problems. The initial examples included learning *foremost* broadcast trees (Casteigts et al. 2011) and *fastest* broadcast trees (Casteigts et al. 2012b) in periodically-varying graphs.

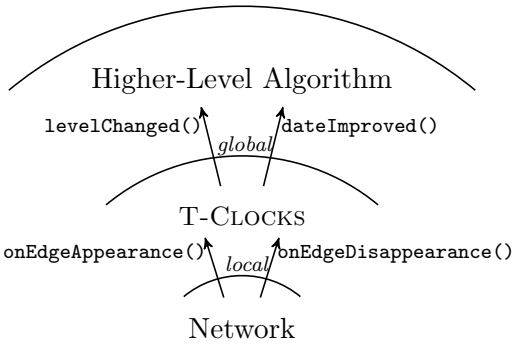


Figure 14: T-Clocks *as an abstraction to track temporal views. Picture from (Casteigts et al. 2012b).*

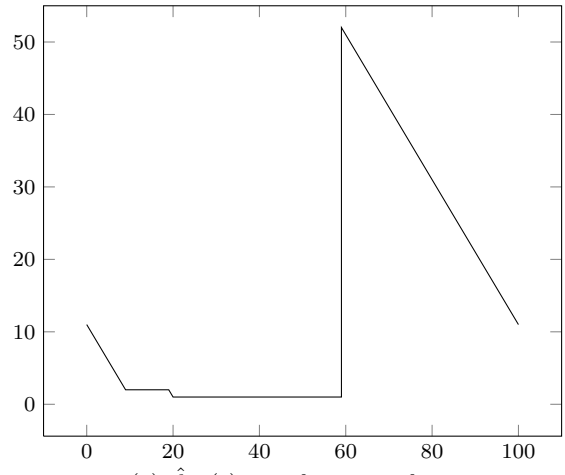
3.3.6 Further concepts

The number of definitions built on top of temporal concepts could grow endlessly, and our aim is certainly not to enumerate all of them. Yet, here is a short list of additional concepts that we believe are general enough to be useful in several analytical contexts.

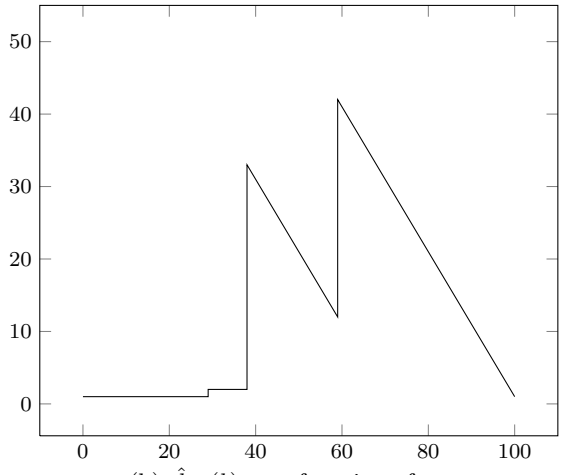
Building on top of the concept of distance, that of *eccentricity* can be similarly mapped into a *topological* and a *temporal* version. The temporal eccentricity of a node u at time t, $\hat{\varepsilon}_t(u)$, is thus defined as

$$\hat{\varepsilon}_t(u) = max\{\hat{d}_{u,t}(v) : v \in V\}; \tag{3}$$

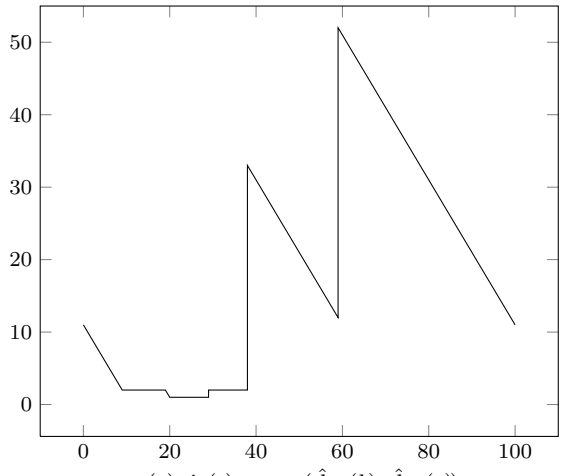
that is, the duration of the "longest" foremost journey from u to any other node. This concept turned out to be very useful in solving the problem of learning fastest broadcast trees in periodically-varying graphs (Casteigts et al. 2012b). Indeed, the best moment to initiate a fastest broadcast is precisely the date of minimal temporal eccentricity for the emitter, which can be determined using T-Clocks (see Section 3.3.5). Figure 15 shows the evolution of the temporal eccentricity of node a in the example graph of Figure 12(a), pictured as the maximum between the two corresponding distances (from a to b and from a to c). Notice that unlike foremost broadcast, which has a different solution for each emission date (the trees are time-dependent), a fastest broadcast tree in a periodic TVG remains optimal regardless of the starting date, since the emitter can simply wait to send the message



(a) $\hat{d}_{a,t}(c)$ as a function of t



(b) $\hat{d}_{a,t}(b)$ as a function of t



(c) $\hat{\varepsilon}_a(t)$=$max(\hat{d}_{a,t}(b), \hat{d}_{a,t}(c))$

Figure 15: *Evolution of the temporal eccentricity of node a in the triangle graph of Figure 12(a) (modulo* 100*). Example from Casteigts et al. (2012b)*

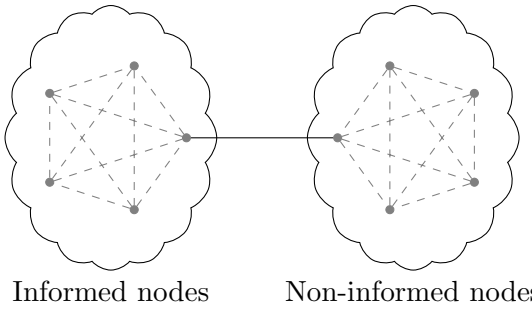


Figure 16: *Small instant diameter vs. large temporal diameter.*

at the time of minimal eccentricity (which holds modulo the period; in this example, any chosen date between 20 and 29).

The concept of diameter can also be mapped into a *topological* and a *temporal* version. The concept of temporal diameter is a time-dependent concept; it is defined at time t as $max\{\hat{\varepsilon}_t(u) : u \in V\}$, or equivalently, as $max\{\hat{d}_t(u,v) : u,v \in V^2\}$. The first temporal versions of eccentricity and diameter seem to have been proposed by Bui-Xuan et al. (2003). Temporal diameter was further investigated from a stochastic point of view by Chaintreau et al. (2008), and under the name "flooding time" by Clementi et al. (2008).

In a recent book chapter, Clementi and Pasquale (2011) it is asked whether it is possible to define an adversarial strategy such that at every time step the snapshot of the graph has a small diameter (say a constant one), and yet the temporal diameter is large (e.g., linear in the number of nodes). This question is left unsolved, inviting the reader to think of it as a little exercise. Figure 16 illustrates a possible configuration that allows us to answer in the affirmative. The adversary only needs to reshuffle the nodes in between every snapshot so as to create a clique of informed nodes and another clique of non-informed nodes, linked by a single edge. This yields a diameter of 3 in every snapshot, while the temporal diameter is $n-1$.

In the same chapter, Clementi and Pasquale introduce a concept of *dynamic expansion*—the dynamic counterpart of the concept of *node expansion* in static graphs—which accounts for the maximal speed of information propagation. Given a subset of nodes $V' \subseteq V$, and two dates $t_1, t_2 \in \mathcal{T}$, the dynamic expansion of V' from time t_1 to time t_2 is the size of the set $\{v \in V \setminus V' : \exists \mathcal{J}_{(u,v)} \in \mathcal{J}^*_{\mathcal{G}[t_1,t_2)} : u \in V'\}$, that is, in a sense, the *collective horizon* of V' in $\mathcal{G}_{[t_1,t_2)}$.

4 Communication and computational models

Distributed algorithms can be formulated in a variety of ways, and using a variety of communication or computational assumptions. This section is a review of the main models and their specifications. For simplicity, the following descriptions assume an undirected graph. We first review low-level models, which into account the communication environment in which the algorithm executes, and then focus on high-level abstraction models that have proved useful in obtaining general results in distributed computation.

4.1 Communication by message passing

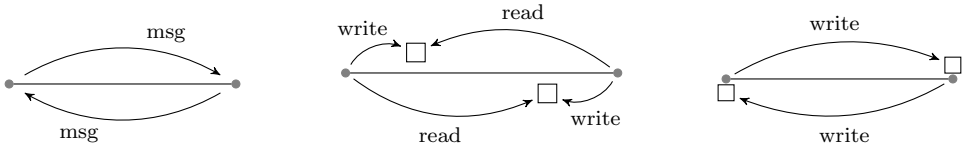
Message passing is by far the most common model in the literature. In this model, nodes communicate by exchanging messages over communication channels that usually reflect the graph structure of the network. The sender node drops a message to be sent in one of its outgoing channels, while the receiver is notified in an event-based fashion when a new message arrives. The model has several variants regarding synchronicity, faults, or the type of medium considered. For example, synchronous message passing consists of *rounded* communication whereby messages sent at the end of round $r-1$ are delivered at destination in the beginning of round r, while asynchronous message passing considers an unpredictable (but in general finite) delivery time. Another important distinction is between a wired communication medium, in which all pairs of neighbours have their own private channel, and a wireless communication medium (more generally called a broadcast medium), in which every message sent by a node is received by all its neighbours. Further variations include cases with or without *port numbering* to distinguish between local neighbours relative to the sending or receiving of a message. The message passing communications model is illustrated in Figure 17(a).

4.2 Communication by registers

In this model, each communication link has two associated registers, one for each direction. Each register can be accessed in read/write mode by one of the two nodes, and in read-only mode by the other. To communicate with a neighbour, a node thus writes in the corresponding register, while the neighbour reads it from time to time in an active way. By reading an outgoing register, an emitter can learn whether the register was accessed (read) by the corresponding neighbour, which allows some sort of synchronicity. The register-based communications model is illustrated in Figure 17(b).

4.3 Communication by mailboxes

In this model, each node has a memory area in which neighbours can write data. A node n_1 wishing to communicate with a neighbour n_2 writes in the memory area of n_2 and, if awaiting a reply, reads the reply in its memory area where n_2 can write. In general, it is assumed that the reading of data by the underlying node deletes it, which is called message *consumption*. The mailbox-based communications model is illustrated in Figure 17(c).



(a) Message passing (b) Registers (c) Mailboxes

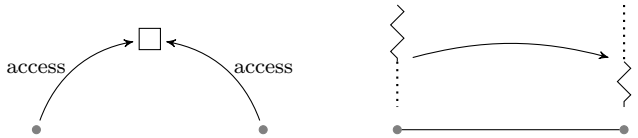
Figure 17: *Communication models.*

4.4 Communication by shared memory

In this model, the different nodes wishing to communicate share a common memory resource. Access to this resource is generally exclusive for writes, but can be read concurrently. Work in this model often faces strong constraints regarding the cost of a central communication system, which constitutes a significant part of the algorithmic efforts and contributions. The shared-memory communications model is illustrated in Figure 18(a).

4.5 Computation by mobile agents

Agents are computational entities that have the ability to migrate and change their execution platform. As such, they can pause the execution of an algorithm, move from a node to another node, then resume this execution. Variants include the possibility or not for the agents to carry data with them. The model of mobile agents was proven computationally equivalent to that of message passing (Chalopin et al. 2006a, Das et al. 2007). Yet, it suggests a programming paradigm that is much different in essence and allows natural thinking for some distributed problems, e.g., map construction (Flocchini et al. 2009a). The mobile agent computational model is illustrated in Figure 18(b).



(a) Shared memory (b) Mobile agents

Figure 18: *Communication models (continued).*

4.6 Abstracting communications

Although a vast majority of algorithms are designed in one of the above models—predominantly the message passing model—the very fact that one of them is chosen implies that the obtained results (e.g., positive or negative characterizations and associated proofs) are limited to the scope of this model. This problem of diversity among formalisms and results, already stated by Lynch (1989) more than twenty years ago, led researchers to consider higher abstractions when studying fundamental properties of distributed systems, some of which we consider in this section.

4.6.1 Local computation by graph relabelling

Local computations by means of *graph relabelling* was proposed as an abstraction by Litovsky et al. (1999). These theoretical tools allow one to represent a distributed algorithm as a set of local interaction rules that are independent from the communications model.

This level of abstraction is achieved by representing the network as a *labelled* graph, where the label of each vertex or edge encodes its algorithmic state. The algorithm is then defined as a set of *relabelling rules* that modify these labels in an atomic way, according to some local pattern.

Figure 19 shows a basic example of algorithm made up of a single rule, $I \overset{0}{-} N \longrightarrow I \overset{1}{=} I$, which builds a spanning tree over an arbitrary graph. Concretely, the effect of the rule is to extend the existing tree by including a connected external node (label N) and the corresponding edge. Initially, all the nodes are labelled N except one distinguished node, the root of the tree, labelled I. All the edges are labelled 0 (they do not belong to the tree). Then, as the rule is applied repeatedly, more nodes and edges are included, leading eventually to a valid spanning tree.

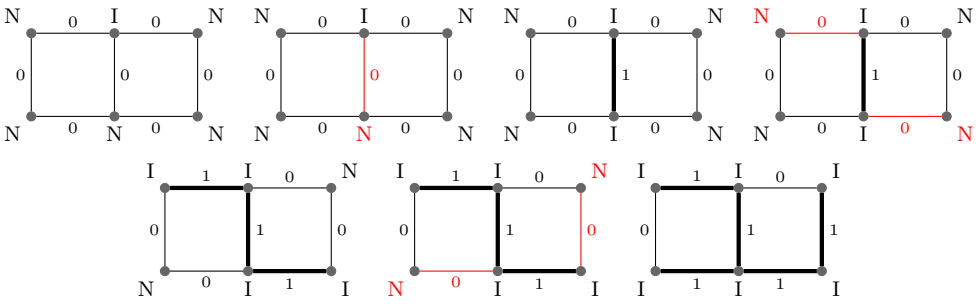


Figure 19: *Spanning tree construction over a static graph, using graph relabelling. Adapted from (Litovsky et al. 1999).*

Several relabelling steps can occur simultaneously, as long as they involve disjoint sets of nodes. The order in which the interactions take place is not specified by the algorithm; it can be regarded as an implementation choice or even an external component (e.g., controlled by an adversary). Thus, from the abstraction level of graph relabelling, several outcomes are possible, e.g., there are several possible spanning trees in our example.

Slightly more formally, let the network topology be represented by a finite undirected loopless graph $G = (V, E)$, with V representing the set of nodes and E representing the set of communication links between them. Let $\lambda : V \cup E \to \mathcal{L}^*$ be a mapping that associates every vertex and edge from G with one or several labels from an alphabet $\mathcal{L}$ (which denotes all the possible states these elements can take). The state of a vertex v or edge e at time t is denoted by $\lambda_t(v)$ or $\lambda_t(e)$, respectively. The whole *labelled graph* is represented by the pair (G, λ).

According to Litovsky et al. (1999), a complete algorithm can be given by a triplet $\{\mathcal{L}, \mathcal{I}, P\}$,

Algorithm 1 A propagation algorithm coded by a single relabelling rule (r_1).

initial states: $\{I, N\}$ (I for the initial emitter, N for all the other vertices)
alphabet: $\{I, N\}$

preconditions(r_1): $\lambda(v_0) = I \wedge \lambda(v_1) = N$

actions(r_1): $\lambda(v_1) := I$

graphical notation :

where $\mathcal{I}$ is the set of initial states, and P is a set of *relabelling rules* (transition patterns) representing the distributed interactions—these rules are considered *uniform* (i.e., same for all nodes). Algorithm 1 shows another example of single-rule algorithm that represents a general broadcasting scheme. We assume here that the label I stands for the state `informed` and N for `non-informed`. Propagating the information thus consists of repeating this rule, starting from the emitter vertex, until all vertices are labelled I. Detecting such a final state is not part of the given algorithm, nor was it for the above spanning tree algorithm. A treatment of termination detection, per se, in the context of relabelling algorithms can be found in (Godard et al. 2002).

The *scope* of computation considered above involved only two neighbour nodes (*pairwise* interaction); however, various scopes of computation can be considered, as depicted in Figure 20.

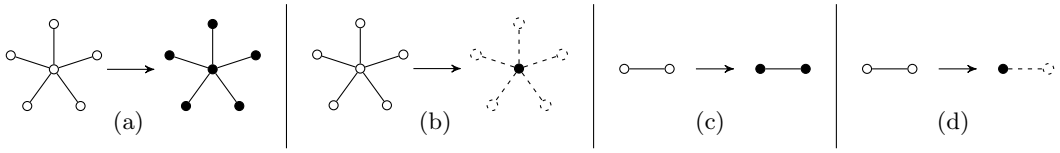


Figure 20: *Different scopes of local computations; the scope of* preconditions *is depicted in white (on the left side of each diagram), while the scope of* actions *is depicted in black (on the right side). The dashed elements represent entities (vertices or edges) that are considered by the preconditions but remain unchanged by the actions.*

One may ask whether the various models in Figure 20 are in fact equivalent in power (e.g., could we simulate any of them by repeating another?). The answer is "no" due to different levels of atomicity (e.g., models 20(a) vs. 20(c)) as well as different levels of symmetry breaking (e.g., models 20(c) vs. 20(d)). The reader is referred the work of Chalopin et al. (2006b) for a detailed hierarchy of these models. Note that equivalences between models would have to be re-considered in a dynamic context, because dynamics may prevent the possibility of applying several steps of a weaker model to simulate a stronger one.

Let us stress that an algorithm does not specify how the nodes synchronize, i.e., how they select each other to perform a common computation step. Dedicated procedures have been designed to fit the various models, e.g., local elections (Métivier et al. 2002) and local rendezvous (Métivier et al. 2003) for star-wise and pairwise interactions, respectively. A direct consequence is that the execution of an algorithm at this level may not be deterministic.

The abstraction offered by local computations makes them powerful tools for studying fundamental properties of distributed algorithms, because it allows to obtain very general results. For example, a negative result (e.g., an impossibility proof or a necessary condition) obtained at this level remains valid within the framework of any communication model (except perhaps quantum ones). Positive results however (e.g., a correctness proof or sufficient conditions) cannot always be transposed into all communication models.

4.6.2 Graph relabelling over dynamic graphs

The graph relabelling formalism was used in a context of dynamic networks by Casteigts et al. (2009), with the aim to prove mathematically that given properties on the dynamics of the network (properties on evolving graphs) are necessary or sufficient conditions to given distributed problems. The formalism itself can be generalized to dynamic graphs by considering new rules that associate dedicated operations to the appearance or disappearance of an incident edge (Casteigts and Chaumette 2005). Figure 21 shows an example algorithm that maintains a forest of spanning trees over a constantly changing delay-tolerant network, trying to minimized the average number of trees per connected component (Casteigts 2006).

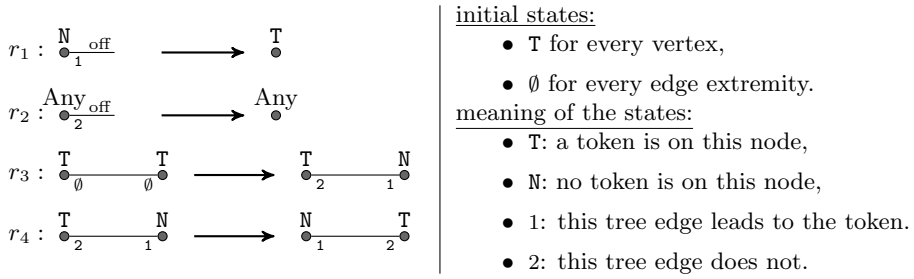


Figure 21: *A spanning forest algorithm based on coalescing and regenerating trees.*

A number of dedicated analytical techniques were developed in the context of graph relabelling systems over static or dynamic graphs; however, these are beyond the scope of the present document.

4.6.3 Population protocols

Another approach that shares the objective of abstracting the communication model is that of *population protocols* suggested by Angluin et al. (2006). Population protocols actually bear a strong resemblance to the pairwise computational model of Figure 20(c). The brilliant idea behind population protocols was to consider the underlying synchronization layer as being representative of node movement (and thus graph dynamics), in such a way that the model is appropriate for the study of certain dynamic networks.

In the original variant, every node is an anonymous finite state machine (FSM) that interacts with other nodes at the same abstraction level as with local computations, i.e., assuming an atomic modification of the state of two neighbouring nodes based on their current state. Interaction takes place between pairs of nodes whenever they meet, which occurs infinitely

often for every pair, according to the initial assumption that the *graph of interaction* is complete (see Figure 22(a)). Weaker assumptions on the graph of interaction have since been considered, such as being only connected, rather than completely connected, or having a tree topology (see Figure 22(b)).

(a) Complete interaction graph (b) Tree-based interaction graph

Figure 22: *Examples of graphs of interaction for population protocols. (An edge represents an interaction that takes place infinitely often.)*

Research around population protocols mainly focuses on examining variations of the initial model and characterizing, for each variant, the type of predicates (formulas of logic related to the nodes states) that could possibly be computed. For example, the initial model was shown to compute the class of *semilinear* predicates, i.e., all predicates definable by first order Presburger arithmetics (Ginsburg and Spanier 1966).

Variants that have been considered include models where nodes or edges are endowed with stronger capabilities, such as

- *Mediated population protocols* (Chatzigiannakis et al. 2010), which consist of the original setting plus a constant size memory available for each edge.
- *Community protocols* (Guerraoui and Ruppert 2009) in which nodes possess unique identifiers and can store a constant number of other identifiers.
- *Passively mobile machines* (Chatzigiannakis et al. 2011) in which nodes are not FSMs anymore, but rather Turing machines with unbounded memory.

Variants also include settings with other interaction schedulers (e.g. having different levels of fairness), or infinite populations (Bournez et al. 2009, Chatzigiannakis and Spirakis 2008).

5 TVG classes

The need to categorize and understand highly dynamic networks led the engineering community to design a variety of *mobility models* for the broadcast/wireless model of communications, where each model captures a particular context by means of movement rules for the nodes. A well-known example is the *random waypoint* model (Guerin 1987), while other specialized models exist for dedicated uses (e.g., for vehicular networks (Harri et al. 2009)). The essential purpose of these models is to be able to reproduce experiments and compare different solutions on a relatively fair basis, thereby providing a common ground for the engineering community to solve practical challenges in highly dynamic networks, e.g., routing and broadcasting (Burgess et al. 2006, Guo and Keshav 2007, Jacquet et al. 2010, Jain et al. 2004, Liu and Wu 2009, Zhang 2006, Zhao et al. 2004).

In the same way as mobility models enables practical investigations in highly dynamic networks, *logical properties* on the graph dynamics, that is, *classes of dynamic graphs*, have the potential to guide a more formal exploration of their analytical aspects. A number of special classes were recently identified, including the following: graphs in which a given set of nodes interact infinitely often (Angluin et al. 2006, 2007, Chatzigiannakis et al. 2009), graphs whose dynamics is unrestricted but which are required to be connected at any instant (Kuhn et al. 2010, O'Dell and Wattenhofer 2005), graphs whose edges appear or disappear with given probabilities (Baumann et al. 2009, Clementi et al. 2008), which have a sufficiently stable root component (Biely et al. 2012), whose schedule is periodic (Casteigts et al. 2011, Flocchini et al. 2012, 2009b, Ilcinkas and Wade 2011), or that guarantee minimal reachability properties (Casteigts et al. 2009).

In this section, we examine some central properties of time-varying graphs and briefly discuss their impact in terms of feasibility and complexity of distributed problems. In doing so, we review and interconnect a large body of literature. In particular, we identify several classes of TVGs defined with respect to basic *properties on the network dynamics*. Some of these classes have been extensively studied in different contexts; e.g., one of them coincides with the family of dynamic graphs over which *population protocols* are defined (see Section 4.6.3). We examine the (strict) inclusion hierarchy among the classes, organized in an ascending order of assumptions—from more general to more specific. This hierarchy was published in a recent paper (Casteigts et al. 2012c). Corresponding to several of the class-defining properties considered here are necessary conditions and impossibility results for basic computations. Thus, the inclusion relationship implies that we can transfer feasibility results (e.g., protocols) to an included class, and impossibility results (e.g., lower bounds) to an including class.

The classes presented in this section are of very different natures: some of them are based on the existence of journeys, some require more classical forms of connectivity (i.e., path-based), and some others are directly defined in terms of edge properties. Some classes allow the network to have a *finite* lifetime, while others are defined over recurrent properties that are assumed to hold infinitely. Finally, some classes are not uniform in the sense that they require properties relative to a subset of nodes only, others are inspired by the fault-tolerance

paradigm (e.g., awaiting for some classical property to be eventually verified). The reader is referred to Table 1 to get a overview of the classes main properties, while going through their definitions.

Class	Journey-based	Path-based	Edge-based	Finite	Uniform	Fault-tolerant
$\mathcal{C}_1$	✓	–	–	✓	–	–
$\mathcal{C}_2$	✓	–	–	✓	–	–
$\mathcal{C}_3$	✓	–	–	✓	✓	–
$\mathcal{C}_4$	✓	–	–	✓	✓	–
$\mathcal{C}_5$	✓	–	–	–	✓	–
$\mathcal{C}_6$	–	–	✓	–	✓	–
$\mathcal{C}_7$	–	–	✓	–	✓	–
$\mathcal{C}_8$	–	–	✓	–	✓	–
$\mathcal{C}_9$	–	✓	–	–	✓	–
$\mathcal{C}_{10}$	–	✓	–	–	✓	–
$\mathcal{C}_{11}$	–	✓	–	–	✓	✓
$\mathcal{C}_{12}$	–	✓	–	–	✓	✓
$\mathcal{C}_{13}$	–	–	✓	–	✓	–

Table 1: *Summary of key properties of the classes.*

5.1 Classes of graphs based on finite properties

The most general class to make sense is perhaps the one wherein at least one node can reach all the others by means of a journey, that is, in the TVG formalism:

Class 1 $\exists u \in V : \forall v \in V, u \rightsquigarrow v$.

This condition is necessary, for example, for broadcast to be feasible at least once from at least one node. This condition is present in all the TVGs we have shown so far, except that of Figure 10 on page 17.

The second class is somewhat uniform:

Class 2 $\exists u \in V : \forall v \in V, v \rightsquigarrow u$.

That is, at least one node can be reached by all the others. An example with this property is node d in the graph of Figure 11 on page 20. This condition is necessary for a variety of tasks, for example for computing a function whose input is distributed over all the nodes (with at least one node generating the output). Any algorithm for which a terminal state must be causally related to all the nodes initial states also falls in this category, including, for example, most leader election or counting algorithms.

Class 3 *(Connectivity over time):* $\forall u, v \in V, u \rightsquigarrow v$.

That is, every node can reach all the others; in other words, the TVG is connected over time (or *temporally* connected). This class contains, for instance, the graph of Figure 9 on page 16, but it does not contain that of Figure 11 on page 20. By the same reasoning as for Class 1 and Class 2, this condition is necessary to enable broadcast from *any* node, to compute a function whose output is known by *all* the nodes, or again to ensure that *every* node has a chance to be elected in a leader election.

These three basic classes show how relations between TVGs properties and the feasibility of algorithms can be *formally* established (Casteigts et al. 2009) based on a combination of evolving graphs and graph relabelling (see also Section 4.6.1). Variants of these classes can be found in recent literature, e.g., (Greve et al. 2011) where the assumption that temporal connectivity is eventually achieved among a stable subset of the nodes is used to implement failure detectors in dynamic networks.

Class 4 *(Round connectivity):* $\forall u, v \in V, \exists \mathcal{J}_1 \in \mathcal{J}^*_{(u,v)}, \exists \mathcal{J}_2 \in \mathcal{J}^*_{(v,u)} : arrival(\mathcal{J}_1) \leq departure(\mathcal{J}_2)$.

That is, every node can reach all the others and be reached back *afterwards*. Most of the graphs seen so far do *not* belong to this class. The graph of Figure 11 on page 20 does so because it contains only two edges whose presence overlap in time. Round connectivity may be required, e.g., for adding explicit termination to broadcast, election, or counting algorithms.

5.2 Classes of graphs based on infinite properties

The classes defined so far are in general relevant in the case that the network lifetime is *finite*, or at least a finite number of topological events are considered. When the lifetime is *infinite*, connectivity over time is generally assumed by default, and more elaborate properties can then be considered.

Class 5 *(Recurrent connectivity):* $\forall u, v \in V, \forall t \in \mathcal{T}, \exists \mathcal{J} \in \mathcal{J}^*_{(u,v)} : departure(\mathcal{J}) > t$.

That is, at any point t in time, the temporal subgraph $\mathcal{G}_{[t,+\infty)}$ remains connected over time. A graph like the one of Figure 12(a) on page 20 belongs to this class because its schedule is periodic and its underlying graph is connected. Both graphs of Figure 22 on page 30 also belong to this class. Recurrent connectivity is implicitly assumed in most works related to delay-tolerant networks. It actually characterizes all those DTNs where routing can always be achieved from any node to any other node, recurrently. This class was referred to as *eventually connected* by Awerbuch and Even (1984), however the terminology

"eventually connected" is also used with different meaning in recent DTNs literature (see below Class 11).

As discussed in Section 3.3.1, the fact that the underlying graph $G = (V, E)$ is connected does not imply that $\mathcal{G}$ is connected over time—the ordering of topological events matters. Such a condition is however *necessary* to allow connectivity over time and thus to perform any type of global computation. For this reason, the following three classes additionally assume that the underlying graph G is connected.

Class 6 *(Recurrence of edges):* $\forall e \in E, \forall t \in \mathcal{T}, \exists t' > t : \rho(e, t') = 1$ *and* G *is connected.*

That is, if an edge appears once, it appears infinitely often (i.e., for any date, there exists a future date when the edge appears again). Examples of graphs in this class include social networks in which we assume two people that interact at some time will eventually interact again. Observe that since the underlying graph G is connected, we have Class 6 $\subseteq$ Class 5. Indeed, if all the edges of a connected graph appear infinitely often, then there must exist, by transitivity, a journey between any pairs of nodes infinitely often.

In a context where connectivity is recurrently achieved, it becomes interesting to look at problems where more specific properties of the journeys are involved, e.g., the possibility to broadcast a piece of information in a shortest, foremost, or fastest manner (see Section 3.3.5 for definitions). Interestingly, these three variants of the same problem have different requirements in terms of TVG properties. It is for example possible to broadcast in a foremost fashion in Class 6, whereas shortest and fastest broadcasts are not possible in this class (Casteigts et al. 2010).

Shortest broadcast becomes possible if the recurrence of edges is bounded in time, and the bound is known to the nodes, a property characterizing the next class:

Class 7 *(Time-bounded recurrence of edges):* $\forall e \in E, \forall t \in \mathcal{T}, \exists t' \in [t, t + \Delta), \rho(e, t') = 1$, *for some* $\Delta \in \mathbb{T}$ *and* G *is connected.*

Examples in this class include professional networks (e.g., in a small company) where persons are supposed to interact with a number of colleagues at least once a week (Δ could here be 12 days, the worst case lag between two interactions). An interesting implication of this class is that the *temporal diameter* becomes bounded by Δ times the *topological diameter* (see definitions in Section 3.3.6). Nodes also have the possibility to learn their neighbourhood in the underlying graph, through waiting a period of Δ (if Δ is known). The feasibility of shortest broadcast follows naturally by using a Δ-rounded breadth-first strategy that minimizes the topological length of journeys.

A particular case of bounded recurrence that is frequently encountered in literature is that of periodic networks:

Class 8 *(Periodicity of edges):* $\forall e \in E, \forall t \in \mathcal{T}, \forall k \in \mathbb{N}, \rho(e,t) = \rho(e,t+kp)$*, for some* $p \in \mathbb{T}$ *and* G *is connected.*

The graph in Figure 12(a) falls into this class. The periodic assumption holds in practice in many cases, including networks whose entities are mobile with periodic movements (satellites, guards tour, subways, or buses). The periodicity assumption within a DTN has been considered in a number of contexts including network exploration (Flocchini et al. 2009b, Ilcinkas and Wade 2011, Flocchini et al. 2012) and routing (Keränen and Ott 2009, Liu and Wu 2009). Periodicity also enables the construction of foremost broadcast trees that can be re-used modulo p for subsequent broadcasts (Casteigts et al. 2011), whereas the more general classes of recurrence requires the use of a different tree for every foremost broadcast.

More generally, the point of exploiting TVG properties is to rely on invariants that are generated by the dynamics (e.g., recurrent existence of journeys, periodic optimality of a broadcast tree, etc.). In some works, particular assumptions on network dynamics are made to obtain invariants of a more classical nature. Below are some examples of classes, formulated using the graph-centric point of view of (discrete-time) evolving graphs, i.e., where $\mathcal{G} = (G, \mathcal{S}_\mathcal{G}, \mathbb{N})$.

Class 9 *(Constant connectivity):* $\forall G_i \in \mathcal{S}_\mathcal{G}, G_i$ *is connected.*

Here, the dynamics of the network are not constrained as long as the graph remains connected in every time step. Examples of this class include sensor networks with sleeping schedules, wherein different subsets of sensors are successively put to sleep to save energy, while a connected subset of them remains awake to serve their purpose. Such a class was used for example by O'Dell and Wattenhofer (2005) to enable progression hypotheses on the broadcast problem. Indeed, if the network is always connected, then at every time step there must exist an edge between an informed node and a non-informed node, which allows one to bound the broadcast time (by $n = |V|$ in the worst case). This class was also considered by Kuhn et al. (2011) for the problem of consensus.

Class 10 *(T-interval connectivity):* $\forall i \in \mathbb{N}, T \in \mathbb{N}, \exists G' \subseteq G : V_{G'} = V_G, G'$ *is connected, and* $\forall j \in [i, i+T-1), G' \subseteq G_j$.

This class is a particular instance of constant connectivity in which a specific spanning connected subgraph of the underlying graph G is available for any period of T consecutive time steps. It was introduced by Kuhn et al. (2010) to study problems such as counting, token dissemination, and computation of functions whose input is spread over all the nodes (considering an adversarial edge schedule). The authors show that computation could be sped up a factor of T compared to the 1-interval connected graphs, that is, graphs of Class 9.

Other classes of TVGs were studied by Ramanathan et al. (2007), which rely on intermediate properties between constant connectivity and connectivity over time. They include Class 11 and Class 12 below.

Class 11 *(Eventual instant-connectivity):* $\forall i \in \mathbb{N}, \exists j \in \mathbb{N} : j \geq i$, G_j *is connected.*

In other words, there is always a future time step in which the network will be instantly connected. This class was simply referred to as *eventual connectivity* in their paper, but since the meaning is different from that used by Awerbuch and Even (1984) (see above, Class 5), we have renamed it to avoid confusion. This class could represent partially DTN networks where the absence of connectivity is only occasional and transient.

Class 12 *(Eventual instant-routability):* $\forall u, v \in V, \forall i \in \mathbb{N}, \exists j \in \mathbb{N} : j \geq i$ *and a path from* u *to* v *exists in* G_j.

That is, for any two nodes, there is always a future time step in which a instant path exists between them. The difference with Class 11 is that these paths may occur at different times for different pairs of nodes. Ramanathan et al. (2007) used Classes 11 and 12 to represent networks where routing protocols for (connected) mobile ad hoc networks eventually succeed if they tolerate transient topological faults.

Most of the works listed above strove to characterize the impact of various temporal properties on problems or algorithms. A reverse approach was considered by Angluin et al. in the field of *population protocols* (Angluin et al. 2006, 2007), where for a given assumption (that any pair of node interacts infinitely often), they characterized all the problems that could be solved in this context. The corresponding class is generally referred to as that of *(complete or otherwise) graph of interaction*.

Class 13 *(Complete graph of interaction): The underlying graph* $G{=}(V, E)$ *is complete, and* $\forall e \in E, \forall t \in \mathcal{T}, \exists t' > t : \rho(e, t'){=}1$.

From a TVG perspective, this class is the specific subset of Class 6, in which the underlying graph G is complete. Various types of schedulers and assumptions were subsequently considered in the field of population protocols, adding further constraints to Class 13 (e.g., weak fairness, strong fairness, bounded, or k-bounded schedulers) as well as interaction graphs which could be less than complete (see also Section 4.6.3).

5.3 Connecting classes into a hierarchy

An interesting aspect of unifying these properties within the same formalism is the possibility to see how they relate to each another, and to compare the various solutions and algorithms that were introduced within. An insight for example can be gained by looking at the short classification shown in Figure 23, where basic relations of inclusion between the above classes are reported. These inclusions are *strict*: for each relation, the parent class contains some TVGs that are not in the child class.

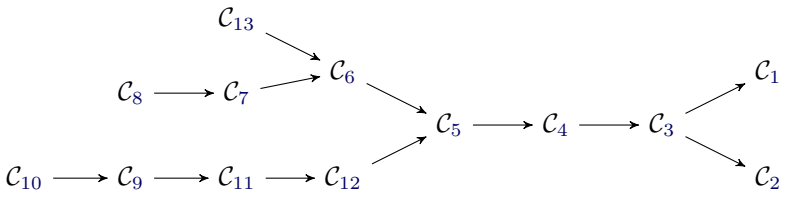


Figure 23: *Relations of inclusion between classes (from specific to general). Picture from (Casteigts et al. 2012c).*

Clearly, one should try to solve a problem in the most general context possible. The rightmost classes are so general that they offer few properties to be exploited by an algorithm, but some intermediate classes, such as Class 5, appear quite central in the hierarchy. This class indeed contains all the classes where significant work was done. A problem solved in this class would therefore apply to virtually all the contexts considered heretofore in the literature.

Such a classification may also be also used to categorize problems themselves. As mentioned above, shortest broadcast is not generally achievable in Class 6, whereas foremost broadcast is. Similarly, it was shown (Casteigts et al. 2010) that fastest broadcast is not feasible in Class 7, whereas shortest broadcast can be achieved with some knowledge. Since Class 7 $\subset$ Class 6, we have

$$\textit{foremostBcast} \preceq \textit{shortestBcast} \preceq \textit{fastestBcast}$$

where $\preceq$ is a partial order on these problems' topological requirements.

This particular area of research is relatively recent and has started gaining importance. As a result, new and more sophisticated classes are now appearing. This is the case, for example, of the class induced by *vertex-stable root components* in directed graphs (Biely et al. 2012), in which the condition is that a subset of nodes (root component) remains strongly connected during a sufficiently long period of time to reach, collectively, all the other nodes in the network. The internal topology of the root component might change as long as they remain strongly connected during the considered period, whose duration is 4 times the temporal diameter of the network. This class is illustrated in Figure 24.

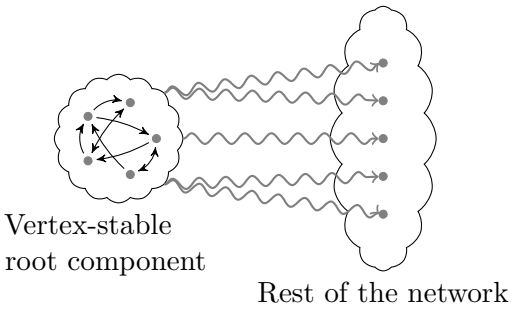


Figure 24: *Vertex-stable root component in directed TVGs (Biely et al. 2012)*

6 Offline analysis

This section is concerned with the *a posteriori* analysis of network traces. We discuss three particular aspects of this general question, which are i) how network traces could be checked for inclusion in some of the classes from Section 5, ii) how temporal concepts can be leveraged to express new phenomenon or properties in complex systems, and iii) how TVGs could be used to study a coarser-grain evolution of network properties, whether these properties are of a classical or a temporal nature (both of which imply different approaches).

6.1 Recognizing TVGs

Let us start with recognition of properties that relate to connectivity. Bhadra and Ferreira (2003) consider the problem of computing connected-component in a given evolving graph $\mathcal{G}$. A *(time-)connected component* in $\mathcal{G}$ is therein defined as a set of nodes $V' \subseteq V$ such that $\forall u, v \in V', u \rightsquigarrow v$. The authors observe that, in general, some of the journeys' edges that contribute to a component may involve nodes *outside* the component, as illustrated in Figure 25. Variations around the concept of connected component include for example *strongly*-connected components (Bhadra and Ferreira 2003), as well as *in*- and *out*-components (Tang et al. 2010a).

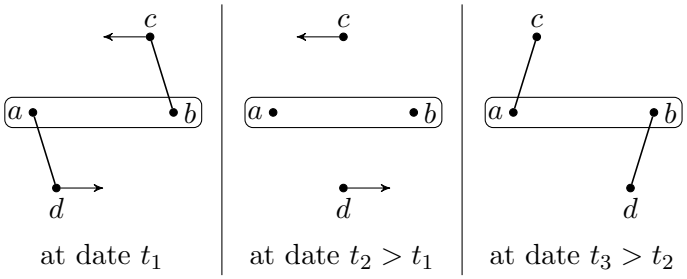


Figure 25: *Example of a connected component (here, $\{a, b\}$ using external edges).*

The problem of determining the largest connected component in a given TVG $\mathcal{G}$ was shown NP-hard by Bhadra and Ferreira (2003), through reduction from the *maximum clique* problem. Yet, checking whether a given set of nodes is a connected component in $\mathcal{G}$ can be done easily, provided a few transformations are made, as described in the same paper. Consider the *transitive closure* of all journeys of a graph $\mathcal{G}$, given as the graph $H = (V, A_H)$, where $A_H = \{(u, v) : u \rightsquigarrow v\}$. The transitive closure is a static and directed graph, as illustrated in Figure 26, since journeys are by nature directed entities. The computation of transitive closures can be done efficiently by building a tree of *shortest* journeys for each node, using any of the algorithms from the paper of Bui-Xuan et al. (2003).

Checking whether a set of nodes is a connected component in $\mathcal{G}$ now becomes the checking of whether it is a clique in H (which is easy). The concept of transitive closure actually allows us to check a graph $\mathcal{G}$ for inclusion in several classes (Casteigts et al. 2009). For instance, the graph is in Class 1 iff H possesses an out-dominating set of size 1; it is in Class 2 iff H possesses an in-dominating set of size 1; and it is in Class 3 iff H is a complete

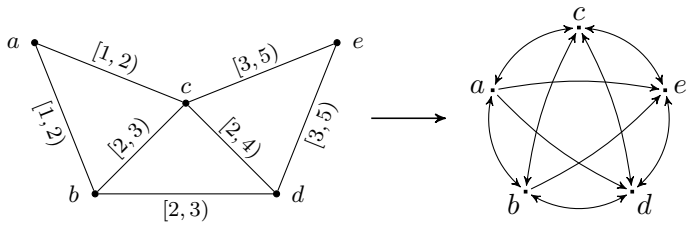


Figure 26: *Transitive closure of journeys. Example from (Casteigts et al. 2009).*

graph. The reader is referred to the above paper for more examples of inclusion checking based on other classes and transformations.

6.2 Transposing the definition of phenomena

We discuss here the possible use of TVGs to express the redefinition (or *translation*) of usual concepts in complex system analysis into a dynamic version. We provide two examples: the *small world* effect, and the *fairness* of a network. Further examples could obviously be found.

6.2.1 Small world

A small-world network is one where the distance between two randomly chosen nodes (in terms of hops) grows logarithmically with the number of nodes in the network. TVG concepts, such as those of journeys, connectivity over time, and temporal distance have been used (Tang et al. 2010b) to characterize the small world behaviour of real-world networks in temporal terms, that is, the fact that there is always a journey of short *duration* between any two nodes. Among the concepts introduced by Tang et al. (2010b) is the *characteristic temporal path length*, defined as

$$\frac{\sum_{u,v \in V} \hat{d}_{t_0}(u,v)}{|V^2|}$$

where t_0 is the first date in the network lifetime $\mathcal{T}$. In other words, this value is the average of temporal distances between all pairs of nodes at starting time. Note that an average of this value over the network lifetime would certainly be meaningful as well.

As per the *topological* meaning (i.e., in terms of hops) of the small world property in a dynamic context, e.g., the fact that "mobile networks have a diameter of 7" (Papadopouli and Schulzrinne 2000), it could be formalized as follows:

$$\forall u, v \in V, \forall t \in \mathcal{T}, \exists \mathcal{J} \in \mathcal{J}^*_{(u,v)} : departure(\mathcal{J}) \geq t, |\mathcal{J}| \leq 7.$$

6.2.2 Fairness and balance

Other properties of interest can take the form of quantities or statistical information. Consider the caricatural example of Figure 27, where nodes a to f represent individuals, each of which meets some other individuals every week (on a periodic basis).

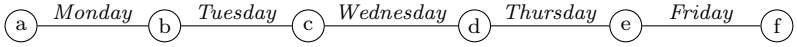


Figure 27: *Weekly interactions between six people.*

A glance at the structure of this network does not reveal any strong anomaly: a) the graph is a line, b) its diameter is 5, c) nodes c and d are more central than the other nodes, etc. However, if we consider the temporal dimension of this graph, it appears that (the interaction described by) the graph is highly unfair and asymmetric: any information originating from a can reach f within 5 to 11 days (depending on what day it is originated), whereas information from f needs about one month to reach a. Node a also appears more central than c and d from a temporal point of view.

We could define here a concept of *fairness* as being the standard deviation among the nodes temporal eccentricities (see Section 3.3.6). This indicator provides an outline on how well the interactions are balanced among nodes. For instance, the TVG of Figure 27 is highly unfair, while the one shown in Figure 28 is fairer (although fairness remains structurally constrained by G, the underlying graph).

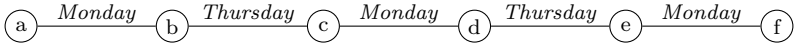


Figure 28: *Weekly interactions between six people—Fairer version.*

A related measure could reflect how balanced the graph is with respect to the time dimension, since the metrics of interest are time-dependent (e.g., the temporal diameter of the TVG in Figure 28 is much lower on Mondays than on Tuesdays). Recent efforts in similar directions include measuring the temporal distance between individuals based on inter-meeting times or e-mail datasets (Kossinets et al. 2008, Kostakos 2009), or the redefinition of further concepts built on top of temporal distance, such as *temporal betweenness* and *temporal closeness*.

6.3 Capturing the coarse-grain evolution

On many occasions in this document, we have focused on the question of how static concepts translate into a dynamic context, e.g., through the redefinition of more basic notions like those of paths (into journeys), distance (into temporal distance) or connectivity (into connectivity over time). From a complex system perspective, these *temporal* indicators, as well as those built on top of them, are completing the set of *atemporal* indicators usually considered, such as (the normal versions of) distance and diameter, density, clustering coefficient, or modularity, to name a few. It is important to keep in mind that all these

indicators, whether temporal or atemporal, essentially account for network properties at a reasonably short time-scale (*fine-grain* dynamics). They do not, however, necessarily reflect how the network evolves over longer periods of time (*coarse-grain* dynamics).

We present below a general approach to looking at the evolution of both atemporal and temporal indicators (Santoro et al. 2011). Looking at the evolution of atemporal indicators can be done by representing the evolution of the network as a sequence of *static* graphs, each of which represents the aggregated interactions over a given time-window. The usual indicators can then be normally measured on these graphs and their evolution studied over time. The case of temporal indicators is more complex because the corresponding evaluation cannot be done on static graphs. The proposed solution is therefore to look at the evolution of temporal indicators through a *sequence* of shorter (and non-aggregated) TVGs, i.e., a sequence of *temporal subgraphs* of the original TVG that cover successive time-windows.

6.3.1 Evolution of atemporal indicators

Atemporal indicators are those whose definition can be stated upon a static graph. They represent structural properties a network has at a given moment of its history.

TVGs as a sequence of static graphs

Given a TVG $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$, one can define the *footprint* of this graph from t_1 to t_2 as the static graph $G^{[t_1,t_2)} = (V, E^{[t_1,t_2)})$ such that $\forall e \in E, e \in E^{[t_1,t_2)} \iff \exists t \in [t_1, t_2), \rho(e, t) = 1$. In other words, the footprint aggregates all interactions of given time windows into static graphs. Let the lifetime $\mathcal{T}$ of the TVG be partitioned into consecutive sub-intervals $\tau = [t_0, t_1), [t_1, t_2) \ldots [t_i, t_{i+1}), \ldots$, where each $[t_k, t_{k+1})$ can be denoted τ_k. We denote the *sequence of footprints* of $\mathcal{G}$ according to τ as the sequence $SF(\tau) = G^{\tau_0}, G^{\tau_1}, \ldots$.

Considering the sequence $SF(\tau)$ with a sufficient interval size allows us to overcome the strong fluctuations of the fine-grain interactions and focus instead on the more general trends in the evolution of the network's structure. Note that the same approach could be considered with a sequence of intervals that are overlapping (i.e., a *sliding* time-window) instead of disjoint ones. Another variation may be considered based on whether the set of nodes in each G^{τ_i} is also allowed to vary.

Since every graph in the sequence is static, any classical network parameter can be directly measured on it. Depending on the parameter and on the application, different choices of granularity are more appropriate to capture meaningful behaviour. At one extreme, each interval could correspond to the smallest time unit (in discrete-time systems), or to the time between any two consecutive modification of the graph; in these cases, the whole sequence becomes equivalent to the model of *evolving graph* (Ferreira 2004). At the other end of the spectrum, i.e., taking $\tau = \mathcal{T}$, the sequence would consist of a single footprint aggregating all interactions over the network lifetime, that is, be equal to G, the underlying graph of $\mathcal{G}$.

Looking at the evolution of atemporal parameters allows to understand how some emerging phenomena occur on the network structure, for instance, the densification of transportation

networks (through diameter or average distance indicators), or the formation of communities in social networks, through *modularity* (Blondel et al. 2008), *cohesion* (Friggeri et al. 2011) or other indicators, e.g., (Alvarez-Hamelin et al. 2008, Börner et al. 2007).

6.3.2 Evolution of temporal indicators

Most temporal concepts—including those mentioned in Section 3.3—are based on replacing the notion of a *path* with that of a *journey*. As a result, they can be decomposed into three versions depending on the type of metric considered (i.e., shortest, foremost, fastest). Since journeys are paths over *time*, the evolution of parameters based on journeys *cannot* be studied using a sequence of aggregated static graphs. For example, there might be a path between x and y in all footprints, and yet possibly no journey between them depending on the precise chronology of interaction. Analyzing the evolution of such parameters requires more than a sequence of static graphs.

TVGs as a sequence of (shorter) TVGs

Temporal subgraphs have been defined in Section 3.3. Roughly speaking, they are themselves TVGs that reproduce all the interactions present in the original TVG for a given time window—without aggregating them. In the same way as for the sequence of footprints, we can now look at the evolution of a TVG through a sequence of shorter TVGs $SF(\tau) = \mathcal{G}^{\tau_0}, \mathcal{G}^{\tau_1}, \ldots$, in which the intervals are either disjoint or overlapping. Looking at the coarse-grain evolution of temporal indicators could allow to answer questions like: how does the temporal distance between nodes evolve over time? Or more generally, how does a network self-organize, optimize, or deteriorate, in terms of temporal efficiency. Using concepts like fairness, defined above, this approach may also help capture the emergence of non-apparent inequalities in a social network.

7 TVGs and formal languages

In this section, we review some recent work that explored the connections between dynamic networks (by means of TVGs), and computability and formal languages. We show how the manipulation of TVGs as automata allowed for a clean characterization of the power of *buffering* in dynamic networks (that is, the ability for a node to store and carry an information for some time before forwarding it). It was shown, in particular, that the expressivity of TVGs drops from that of a Turing machine to that of a finite state machine (FSM) when unrestricted buffering is allowed for the nodes. We then show how a TVG (in the first case) can be used to simulate a Turing machine, which complements these results in constructive way.

7.1 TVGs seen as automata

Given a dynamic network modelled as a TVG $\mathcal{G}$, a journey in $\mathcal{G}$ can be viewed as a word in the alphabet of the edge labels. In this light, the class of feasible journeys defines the language $L_f(\mathcal{G})$ expressed by $\mathcal{G}$, where $f \in \{wait, nowait\}$ indicates whether or not indirect journeys are considered feasible by the environment. Hence, a TVG $\mathcal{G}$ whose edges are labelled over Σ, can be viewed as a TVG-automaton $\mathcal{A}(\mathcal{G}) = (\Sigma, S, I, \mathcal{E}, F)$ where Σ is the input *alphabet*; $S = V$ is the set of *states*; $I \subseteq S$ is the set of *initial states*; $F \subseteq S$ is the set of *accepting states*; and $\mathcal{E} \subseteq S \times \mathcal{T} \times \Sigma \times S \times \mathcal{T}$ is the set of *transitions* such that $(s, t, a, s', t') \in \mathcal{E}$ iff $\exists e = (s, s', a) \in E : \rho(e, t) = 1, \zeta(e, t) = t' - t$.

Figure 29 shows an example of a deterministic TVG-automaton $\mathcal{A}(\mathcal{G})$ that recognizes the context-free language $a^n b^n$ for $n \geq 1$ (using only direct journeys). The presence and latency of the edges of $\mathcal{G}$ are specified in Table 2, where p and q are two distinct prime numbers greater than 1, v_0 is the initial state, v_2 is the accepting state, and reading starts at time $t = 1$.

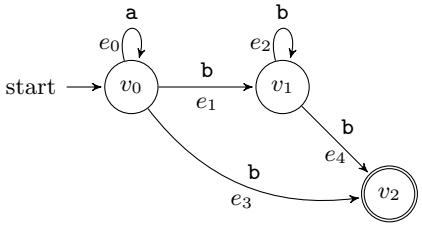


Figure 29: *A TVG-automaton $\mathcal{A}(\mathcal{G})$ such that $L_{nowait}(\mathcal{G}) = \{a^n b^n : n \geq 1\}$. Picture from (Casteigts et al. 2012a).*

The reader may have noticed the basic principle employed here (and in the other examples from the same work) consist in using latencies as a means to *encode* words into time, and presences as a means to *select* through opening the appropriate edges at the appropriate time.

In (Casteigts et al. 2012a) this model was used to study the difference of expressivity

e	Presence $\rho(e,t) = 1$ iff		Latency $\zeta(e,t)$ =
e_0	always true		$(p-1)t$
e_1	$t > p$		$(q-1)t$
e_2	$t \neq p^i q^{i-1}$,$i > 1$		$(q-1)t$
e_3	$t = p$		any
e_4	$t = p^i q^{i-1}$, $i > 1$		any

Table 2: *Presence and latency functions for the edges of $\mathcal{G}$.*

between TVGs in which direct journeys are the only type of journey feasible, and TVGs where indirect journeys are also allowed, i.e., nodes can buffer .

The authors focused on the sets of languages $\mathcal{L}_{nowait} = \{L_{nowait}(\mathcal{G}) : \mathcal{G} \in \mathcal{U}\}$ and $\mathcal{L}_{wait} = \{L_{wait}(\mathcal{G}) : \mathcal{G} \in \mathcal{U}\}$, where $\mathcal{U}$ is the set of all TVGs; that is, the languages expressed when waiting is or is not allowed. For each of these two sets, the complexity of recognizing any language in the set (that is, the computational power needed by the accepting automaton) defines the level of difficulty of the environment. The authors first study the expressivity of TVGs when waiting is *not* allowed—that is, the only feasible journeys are direct ones—and prove that the set $\mathcal{L}_{nowait}$ contains all computable languages. The proof is constructive.

The authors next examine the expressivity of TVGs if indirect journeys are allowed—that is, entities have the choice to wait for future opportunities of interaction rather than seizing only those that are directly available. In striking contrast with the non-waiting case, the languages $\mathcal{L}_{wait}$ recognized by TVG-automata are precisely the set of *regular* languages. In other words, when waiting is no longer forbidden, the power of the accepting automaton (i.e., the difficulty of the environment, the power of the adversary), drops drastically from being at least as powerful as a Turing machine, to becoming that of a finite state machine. This large gap gives a measure of the computational power of waiting. The gap is indeed large; informally speaking, Turing machines (TMs) capture the computational power of today's computers, without considering performance or memory capacity, whereas the amount of things a FSMs can do is very limited (e.g., it cannot check whether a given word "$aa \ldots aabb \ldots bb$" has the same number of as and bs).

To better understand the power of waiting, the authors then turn their attention to *bounded waiting*—that is, when indirect journeys are considered feasible if the pause between consecutive edges in the journeys have a bounded duration $d > 0$. In other words, at each step of the journey, waiting is allowed only for at most d time units. They examine the set $\mathcal{L}_{wait[d]}$ of the languages expressed by TVGs in this case and prove the negative result that the complexity of the environment is not affected by allowing waiting for a limited amount of time, that is, for any fixed $d \geq 0$, $\mathcal{L}_{wait[d]} = \mathcal{L}_{nowait}$. As a result, the power of the adversary is decreased only if it has no control over the length of waiting, i.e., if the waiting is unpredictable.

7.2 Simulation of a Turing machine

If the nodes have no ability to wait, it is then possible to encode the behavior of a Turing machine (TM) into the direct journeys of a TVG. The general principle is similar to the construction of TVG-automata (see Section 7.1) in the sense that the nodes of the TVG corresponds to the states of the TM and edges correspond to the TM internal transitions. The difficulty here is to represent (encode) the state of the TM *tape* (i.e., its working memory), which is done by means of time itself.

The basic idea is to encode the state of the tape as the *decimal* part of the current time by means of a base 3 number that includes the binary content of the tape, plus three "2"s representing respectively the beginning of the tape, the current position of the tape head, and the end of the tape. For example, tape 010011010 with head on the next-to-last 0 would be encoded as 0.201001120102, which gives 0.375122 in base 10. The latency and presence functions of the TVG can then be tuned to simulate the reading or writing of this state, respectively, by enabling the presence function at this particular time (reading), or setting the transition latency to the corresponding duration (writing), these dates holding modulo 1. This is illustrated in Figure 30.

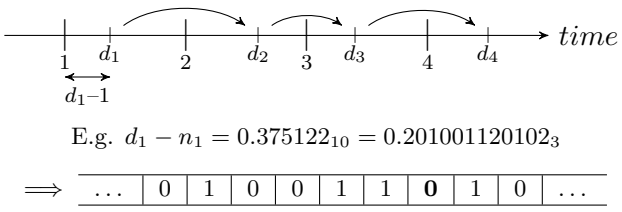


Figure 30: *Simulation of a Turing machine by a time-varying graph*

8 Random TVGs

In this report, we are mainly concerned with deterministic aspects of dynamic networks and related algorithms. As such, we are not considering the role of randomness as central. However, important results related to randomness have been obtained the past few years that would be unfair ignore here. Furthermore, some of these results shed light on general aspects of dynamic networks that hold besides the stochastic context (e.g., transition phases in temporal diameter).

Randomness in TVGs can be introduced at several different levels. The most direct one is clearly that provided by *probabilistic* TVGs, where the presence function $\rho : E \times \mathcal{T} \to [0, 1]$ indicates the *probability* that a given edge is available at a given time. In a context of mobility, the probability distribution of ρ is intrinsically related to the expected mobility of the nodes. Popular examples of random mobility models include the *random waypoint* and *random direction* models (Camp et al. 2002), where waypoints of consecutive movements are chosen uniformly at random. Mobility models from social networks include the time-variant community model (Hsu et al. 2007), and the more recent home-cell community-based mobility model (Boldrini and Passarella 2010).

Definitions of *random* TVG differ depending on whether time is discrete or continuous. A *(discrete-time) random* TVG is a TVG whose lifetime is an interval of $\mathbb{N}$ and whose *sequence of characteristic graphs* $\mathcal{S}_\mathcal{G} = G_1, G_2, \ldots$ is such that every G_i is an Erdös and Rényi random graph, that is, $\forall e \in V^2, \mathbb{P}[e \in E_{G_i}] = p$ for some p; this definition was introduced by Chaintreau et al. (2008).

One peculiarity of discrete-time random TVGs is that the G_is are independent with respect to each other. While this definition allows purely random graphs, it does not capture some properties of real world networks, such as the fact that an edge may be more likely to be present in G_{i+1} if it is already present in G_i. This question is addressed by Clementi et al. (2008) by introducing *edge-Markovian evolving graphs*. These are discrete-time evolving graphs in which the presence of every edge follows an individual Markovian process. More precisely, the sequence of characteristic graph $\mathcal{S}_\mathcal{G} = G_1, G_2, ..$ is such that

$$\begin{cases} \mathbb{P}[e \in E_{G_{i+1}} | e \notin E_{G_i}] = p \\ \mathbb{P}[e \notin E_{G_{i+1}} | e \in E_{G_i}] = q \end{cases}$$

for some p and q called *birth rate* and *death rate*, respectively. The probability that a given edge remains absent or present from G_i to G_{i+1} is obtained by the complement of p and q. The very idea of considering a Markovian evolving graph seems to come from Avin et al. (2008), in which the authors consider a particular case that is substantially equivalent to the discrete-time random TVG (Chaintreau et al. 2008). Variations around the model of edge-markovian evolving graphs include cases where G_{i+1} depends not only on G_i, but also on older graphs G_{i-1}, $G_{i-2}, \ldots$ (the edges follow a higher order Markovian process) (Grindrod and Parsons 2010). Edge-Markovian evolving graphs were used by Clementi et al.

(2008), along with the concept of dynamic expansion (see Section 3.3.6) to address stochastic questions such as Does dynamics necessarily slow down a broadcast? or Can random node mobility be exploited to speed-up information spreading? Baumann et al. (2009) extended this work in by establishing tight bounds on the propagation time for any birth and death rates.

A *continuous-time random* TVG is a TVG in which the appearance of every edge obeys a Poisson process, that is, $\forall e \in V^2, \forall t_i \in App(e), \mathbb{P}[t_{i+1} - t_i < d] = \lambda e^{\lambda x}$ for some λ; this definition is introduced by Chaintreau et al. (2008). (It is interesting to note that in their definition of random TVG, the authors rely on a graph-centric point of view in discrete time and on an edge-centric point of view in continuous time. The same trend can actually be observed in most of the work we reviewed here.)

Random TVGs, both discrete-time and continuous-time, were used to characterize phase transitions between no-connectivity and connectivity over time as a function of the number of nodes, a given time-window duration, and constraints on both the topological and temporal lengths of journeys (Chaintreau et al. 2008).

9 Conclusion

In this report, we reviewed a collection of models and formalisms that are helpful in designing and analyzing deterministic algorithms in dynamic networks. At the centre lies the formalism of time-varying graphs (or evolving graphs), based on which a range of temporal concepts can be elegantly formulated. We showed how these tools can be leveraged to unify previous efforts and identify properties of dynamic networks that impact the feasibility or complexity of algorithms. Such properties could be, for instance, that a temporal path (or journey) exist between some nodes over the lifetime of the network, or that every snapshot of the dynamic network, taken individually, remains connected in a classical way.

Based on the new concepts and graph properties, we presented a hierarchy of thirteen classes of dynamic networks that are related to each other by means of inclusion relations. Each of the classes was shown, in the recent literature, to bear a relation to a specific problem, such as *election*, *spanning trees*, *broadcast*, *routing*, or *consensus*. These relations are intrinsically related to the concern of finding deterministic solutions. Indeed, a class is typically defined by temporal properties that can be proven necessary or sufficient for the successful execution of a given algorithm.

Given a class of dynamic graph, a central question is to know what real-world mobility context this class represents, or put differently, what mobility context would generate network traces that belong to this class. The relevance of this question goes both ways: given a real-world mobility context, what class (or classes) of dynamic graph contain all the network instances such a context yield. We discussed some elements of answer for these questions. In particular, we provided automated ways to check for the inclusion of a dynamic graph into some of the classes.

Finding connections between real-world mobility and dynamic graph properties is appealing in several respects. As stated in the introduction, most dynamic networks exhibit chaotic features and might even look random at first sight, but in the vast majority of the cases there actually exists some form of structure underlying the dynamics. Finding the relations between real-world mobility and graph properties has the potential to reveal such structure, and as a result, to indicate what deterministic algorithms could run, or not, in a given mobility context.

Even though dynamics generally result from the movement of nodes, we have seen that it could also result from interference, or even communications scheduling among static nodes. While discussing the main physical abstraction models, we pointed out, in particular, how the SINR model is likely to induce dynamic topologies regardless of the nodes movements, due to the variations in the communication regions that are based on interference. It is, however, likely that such dynamics are very specific, and therefore exploitable from an algorithmic standpoint. On a similar note, the pairwise interaction caused by most computational models (such as population protocols or graph relabelling) could be seen as yet another level of dynamics. This level should be characterized, again, in terms of dynamic graph properties that have implications on the feasibility or complexity of deterministic solutions.

The concepts, models, and formalisms presented in this report are intended to serve as a basis for more elaborate theoretical tools, directed towards the design and analysis of distributed algorithms in dynamic networks. An upcoming report (Casteigts and Flocchini 2013, CR 2013-021), which is part of the contract, reviews some of these tools, together with concrete examples of algorithms and analyses around most classical distributed problems. Another important topic to be discussed in how new temporal concepts in graph theory impact the very definition of network tasks, yielding several variants for each original problem.

References

Alicherry, M., Bhatia, R., and Li, L. (2005), Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks, In *Proceedings of the 11th annual international conference on mobile computing and networking (MOBICOM)*, pp. 58–72, ACM.

Alvarez-Hamelin, J., Dall'Asta, L., Barrat, A., and Vespignani, A. (2008), K-core decomposition of Internet graphs: hierarchies, self-similarity and measurement biases, *Networks and Heterogeneous Media*, 3, 371.

Andrews, M. and Dinitz, M. (2009), Maximizing capacity in arbitrary wireless networks in the SINR model: Complexity and game theory, In *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM)*, pp. 1332–1340, IEEE.

Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M., and Peralta, R. (2006), Computation in networks of passively mobile finite-state sensors, *Distributed Computing*, 18(4), 235–253.

Angluin, D., Aspnes, J., Eisenstat, D., and Ruppert, E. (2007), The computational power of population protocols, *Distributed Computing*, 20(4), 279–304.

Avin, C., Emek, Y., Kantor, E., Lotker, Z., Peleg, D., and Roditty, L. (2009), SINR diagrams: towards algorithmically usable SINR models of wireless networks, In *Proceedings of the 28th ACM symposium on principles of distributed computing (PODC)*, pp. 200–209, ACM.

Avin, C., Koucky, M., and Lotker, Z. (2008), How to explore a fast-changing world, In *Proc. 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 121–132, Reykjavik, Iceland: Springer.

Awerbuch, B. and Even, S. (1984), Efficient and reliable broadcast is achievable in an eventually connected network, In *Proceedings of the 3rd ACM symposium on Principles of distributed computing (PODC)*, pp. 278–281, Vancouver, Canada: ACM.

Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X. (2006), Group formation in large social networks: membership, growth, and evolution, In *Proc. 12th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 44–54.

Balasubramanian, A., Zhou, Y., Croft, B., Levine, B., and Venkataramani, A. (2007), Web search from a bus, In *Proc. 2nd ACM Workshop on Challenged Networks (CHANTS)*, pp. 59–66.

Balasundaram, B. and Butenko, S. (2008), Optimization problems in unit-disk graphs, *Encyclopedia of optimization*, pp. 2832–2844.

Barrière, L., Fraigniaud, P., and Narayanan, L. (2001), Robust position-based routing in wireless ad hoc networks with unstable transmission ranges, In *Proceedings of the 5th international workshop on discrete algorithms and methods for mobile computing and communications (DIAL-M)*, pp. 19–27, ACM.

Baumann, H., Crescenzi, P., and Fraigniaud, P. (2009), Parsimonious flooding in dynamic graphs, In *Proceedings of the 28th ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 260–269, Calgary, Canada: ACM.

Berman, K. (1996), Vulnerability of scheduled networks and a generalization of Menger's Theorem, *Networks*, 28(3), 125–134.

Bhadra, S. and Ferreira, A. (2003), Complexity of Connected Components in Evolving Graphs and the Computation of Multicast Trees in Dynamic Networks, In *Proceedings of the 2nd International Conference on Ad Hoc, Mobile and Wireless Networks (AdHoc-Now)*, pp. 259–270, Montreal, Canada: Springer.

Biely, M., Robinson, P., and Schmid, U. (2012), Agreement in Directed Dynamic Networks, In *Proceedings of the 19th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*.

Blondel, V., Guillaume, J., Lambiotte, R., and Lefebvre, E. (2008), Fast unfolding of communities in large networks, *Journal of Statistical Mechanics: Theory and Experiment*, 2008, P10008.

Boldrini, C. and Passarella, A. (2010), HCMM: Modelling spatial and temporal properties of human mobility driven by users social relationships, *Comput. Commun.* to appear, doi:10.1016/j.comcom.2010.01.013.

Börner, K., Sanyal, S., and Vespignani, A. (2007), Network science, *Annual review of information science and technology*, 41(1), 537–607.

Bose, P., Morin, P., Stojmenović, I., and Urrutia, J. (1999), Routing with guaranteed delivery in ad hoc wireless networks, In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications (DIAL-M)*, pp. 48–55, ACM.

Bournez, O., Chassaing, P., Cohen, J., Gerin, L., and Koegler, X. (2009), On the convergence of population protocols when population goes to infinity, *Applied Mathematics and Computation*, 215(4), 1340–1350.

Breu, H. and Kirkpatrick, D. (1998), Unit disk graph recognition is NP-hard, *Computational Geometry*, 9(12), 3–24.

Bui-Xuan, B., Ferreira, A., and Jarry, A. (2003), Computing shortest, fastest, and foremost journeys in dynamic networks, *International Journal of Foundations of Computer Science*, 14(2), 267–285.

Burgess, J., Gallagher, B., Jensen, D., and Levine, B. (2006), Maxprop: Routing for vehicle-based disruption-tolerant networks, In *Proceedings of the 25th IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–11, Barcelona, Spain: IEEE.

Camp, T., Boleng, J., and Davies, V. (2002), Diameter of the world-wide web, *Wireless Communications and Mobile Computing.*, 2(5), 483–502.

Cardei, I., Liu, C., and Wu, J. (2007), *Routing in Wireless Networks with Intermittent Connectivity*.

Casteigts, A. (2006), Model Driven capabilities of the DA-GRS model, In *Proceedings of the 1st International Conference on Autonomic and Autonomous Systems (ICAS'06)*, pp. 24–32, Washington, DC, USA: IEEE Computer Society.

Casteigts, A. and Chaumette, S. (2005), Dynamicity-Aware Graph Relabeling Systems (DA-GRS), a local computation-based model to describe MANet algorithms, In *Proceedings of the 17th Conference on Parallel and Distributed Computing and Systems (PDCS'05)*, pp. 231–236, Dallas, USA.

Casteigts, A., Chaumette, S., and Ferreira, A. (2009), Characterizing Topological Assumptions of Distributed Algorithms in Dynamic Networks, In *Proceedings of the 16th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pp. 126–140, Piran, Slovenia: Springer. (Full version on *arXiv:1102.5529*).

Casteigts, A. and Flocchini, P. (2013), Deterministic Algorithms in Dynamic Networks: Problems, analysis, and algorithmic tools, (CR 2013-021) Defence R&D Canada – Ottawa.

Casteigts, A., Flocchini, P., Godard, E., Santoro, N., and Yamashita, M. (2012), Expressivity of Time-Varying Graphs and the Power of Waiting in Dynamic Networks, *CoRR*, Vol. abs/1205.1975.

Casteigts, A., Flocchini, P., Mans, B., and Santoro, N. (2010), Deterministic Computations in Time-Varying Graphs: Broadcasting under Unstructured Mobility, In *Proceedings of 5th IFIP Conference on Theoretical Computer Science (TCS)*, pp. 111–124, Brisbane, Australia: Springer.

Casteigts, A., Flocchini, P., Mans, B., and Santoro, N. (2011), Measuring Temporal Lags in Delay-Tolerant Networks, In *25th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, pp. 209–218, Anchorage, USA: IEEE.

Casteigts, A., Flocchini, P., Mans, B., and Santoro, N. (2012), Measuring Temporal Lags in Delay-Tolerant Networks, *IEEE Transactions on Computers*. In press.

Casteigts, A., Flocchini, P., Quattrociocchi, W., and Santoro, N. (2012), Time-varying graphs and dynamic networks, *International Journal of Parallel, Emergent and Distributed Systems*, 27(5), 387–408.

Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., and Scott, J. (2007), Impact of human mobility on opportunistic forwarding algorithms, *IEEE Transactions on Mobile Computing*, 6(6), 606–620.

Chaintreau, A., Mtibaa, A., Massoulie, L., and Diot, C. (2008), The diameter of opportunistic mobile networks, *Communications Surveys & Tutorials*, 10(3), 74–88.

Chalopin, J., Godard, E., Métivier, Y., and Ossamy, R. (2006), Mobile agent algorithms versus message passing algorithms, *Principles of Distributed Systems*, pp. 187–201.

Chalopin, J., Métivier, Y., and Zielonka, W. (2006), Local computations in graphs: the case of cellular edge local computations, *Fundamenta Informaticae*, 74(1), 85–114.

Chatzigiannakis, I., Michail, O., Nikolaou, S., Pavlogiannis, A., and Spirakis, P. (2010), All symmetric predicates in NSPACE (n 2) are stably computable by the mediated population protocol model, In *Proceedings of the 35th Symposium on Mathematical Foundations of Computer Science (MFCS)*, pp. 270–281, Springer.

Chatzigiannakis, I., Michail, O., Nikolaou, S., Pavlogiannis, A., and Spirakis, P. (2011), Passively mobile communicating machines that use restricted space, *Theoretical Computer Science*, 412(46), 6469–6483.

Chatzigiannakis, I., Michail, O., and Spirakis, P. (2009), Mediated population protocols, *36th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 363–374.

Chatzigiannakis, I. and Spirakis, P. (2008), The dynamics of probabilistic population protocols, In *Proceedings of the 22nd International Symposium on Distributed Computing (DISC)*, pp. 498–499, Springer.

Clementi, A., Macci, C., Monti, A., Pasquale, F., and Silvestri, R. (2008), Flooding time in edge-markovian dynamic graphs, In *Proceedings of the 27th ACM Symposium on Principles of distributed computing (PODC)*, pp. 213–222, Toronto, Canada: ACM.

Clementi, A. and Pasquale, F. (2011), Information Spreading in Dynamic Networks: An Analytical Approach, In Nikoletseas, S. and Rolim, J., (Eds.), *Theoretical Aspects of Distributed Computing in Sensor Networks*, Ch. 19, pp. 591–619, Springer.

Das, S., Flocchini, P., Santoro, N., and Yamashita, M. (2007), Fault-tolerant simulation of message-passing algorithms by mobile agents, *Structural Information and Communication Complexity*, pp. 289–303.

Díaz, J., Mitsche, D., and Pérez, X. (2009), On the probability of the existence of fixed-size components in random geometric graphs, *Advances in Applied Probability*, 41(2), 344–357.

Díaz, J., Mitsche, D., and Santi, P. (2011), Theoretical aspects of graph models for MANETs, *Theoretical aspects of distributed computing in sensor networks*, pp. 161–190.

Eagle, N. and (Sandy) Pentland, A. (2006), Reality mining: sensing complex social systems, *Personal Ubiquitous Comput.*, 10(4), 255–268.

Erdős, P. and Rényi, A. (1960), On the evolution of random graphs, Akad. Kiadó.

Eriksson, J., Balakrishnan, H., and Madden, S. (2008), Cabernet: Vehicular Content Delivery Using WiFi, In *Proceedings of the 14th ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 199–210, San Francisco, USA: ACM.

Ferreira, A. (2004), Building a reference combinatorial model for MANETs, *IEEE Network*, 18(5), 24–29.

Flocchini, P., Kellett, M., Mason, P., and Santoro, N. (2009), Map construction and exploration by mobile agents scattered in a dangerous network, In *IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*, pp. 1–10, IEEE.

Flocchini, P., Kellett, M., Mason, P., and Santoro, N. (2012), Searching for black holes in subways, *Theory of Computing Systems*, 50(1), 158–184.

Flocchini, P., Mans, B., and Santoro, N. (2009), Exploration of periodically varying graphs, In *Proceedings of 20th International Symposium on Algorithms and Computation (ISAAC)*, pp. 534–543.

Floriano, P., Goldman, A., and Arantes, L. (2011), Formalization of the necessary and sufficient connectivity conditions to the distributed mutual exclusion problem in dynamic networks, In *10th IEEE International Symposium on Network Computing and Applications (NCA)*, pp. 203–210, IEEE.

Friggeri, A., Chelius, G., and Fleury, E. (2011), Egomunities, Exploring Socially Cohesive Person-based Communities, *Arxiv preprint arXiv:1102.2623*.

Gilbert, E. (1961), Random plane networks, *Journal of the Society for Industrial and Applied Mathematics*, 9(4), 533–543.

Ginsburg, S. and Spanier, E. (1966), Semigroups, Presburger formulas and languages, *Pacific Journal of Mathematics*, 16(2), 285–296.

Godard, E., Métivier, Y., Mosbah, M., and Sellami, A. (2002), Termination detection of distributed algorithms by graph relabelling systems, *Graph Transformation*, pp. 106–119.

Greve, F., Arantes, L., and Sens, P. (2011), What model and what conditions to implement unreliable failure detectors in dynamic networks?, In *Proceedings of the 3rd International Workshop on Theoretical Aspects of Dynamic Distributed Systems*, pp. 13–17, Rome, Italy: Springer.

Grindrod, P. and Parsons, M. (2010), Social networks: Evolving graphs with memory dependent edges, Technical Report MPS_2010-02, University of Reading.

Grossglauser, M. and Tse, D. (2002), Mobility increases the capacity of ad hoc wireless networks, *IEEE/ACM Transactions on Networks*, 10(4), 477–486.

Grossglauser, M. and Vetterli, M. (2003), Locating nodes with EASE: Last encounter routing in ad hoc networks through mobility diffusion, In *Proceedings of 22nd Conference on Computer Communications (INFOCOM)*, Vol. 3, pp. 1954–1964, San Francisco, USA: IEEE.

Guerin, R. (1987), Channel occupancy time distribution in a cellular radio system, *IEEE Transactions on Vehicular Technology*, 36(3), 89–99.

Guerraoui, R. and Ruppert, E. (2009), Names Trump Malice: Tiny Mobile Agents Can Tolerate Byzantine Failures, In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 484–495, Springer-Verlag.

Guo, S. and Keshav, S. (2007), Fair and efficient scheduling in data ferrying networks, In *Proceedings of ACM Conference on Emerging Network Experiment and Technology (CoNEXT)*, New York, USA: ACM.

Gupta, P. and Kumar, P. (1998), Critical power for asymptotic connectivity, In *Proceedings of the 37th IEEE Conference on decision and control (CDC)*, Vol. 1, pp. 1106–1110, IEEE.

Gupta, P. and Kumar, P. (2000), The capacity of wireless networks, *IEEE Transactions on Information Theory*, 46(2), 388–404.

Harary, F. and Gupta, G. (1997), Dynamic graph models, *Mathematical and Computer Modelling*, 25(7), 79–88.

Harri, J., Filali, F., and Bonnet, C. (2009), Mobility models for vehicular ad hoc networks: a survey and taxonomy, *IEEE Communications Surveys & Tutorials*, 11(4), 19–41.

Holme, P. (2005), Network reachability of real-world contact sequences, *Physical Review E*, 71(4), 46119.

Hsu, W., Spyropoulos, T., Psounis, K., and Helmy, A. (2007), Modeling time-variant user mobility in wireless mobile networks, In *Proceedings of 26th IEEE Conference on Computer Communications (INFOCOM)*, pp. 758–766, Anchorage, USA: IEEE.

Ilcinkas, D. and Wade, A. (2011), On the Power of Waiting when Exploring Public Transportation Systems, *Proceedings of the 15th International Conference on Principles of Distributed Systems (OPODIS)*, pp. 451–464.

Jacquet, P., Mans, B., and Rodolakis, G. (2010), Information propagation speed in mobile and delay tolerant networks, *IEEE Transactions on Information Theory*, 56(1), 5001–5015.

Jain, K., Padhye, J., Padmanabhan, V., and Qiu, L. (2005), Impact of interference on multi-hop wireless network performance, *Wireless networks*, 11(4), 471–487.

Jain, S., Fall, K., and Patra, R. (2004), Routing in a delay tolerant network, In *Proceedings of Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, pp. 145–158.

Jones, E., Li, L., Schmidtke, J., and Ward, P. (2007), Practical routing in delay-tolerant networks, *IEEE Transactions on Mobile Computing*, 6(8), 943–959.

Kellett, M. (2012), Black hole search in the network and subway models, Ph.D. thesis, University of Ottawa.

Kempe, D. and Kleinberg, J. (2002), Protocols and impossibility results for gossip-based communication mechanisms, In *Proceedings of 43rd Symposium on Foundations of Computer Science (FOCS)*, pp. 471–480, Vancouver, USA: IEEE.

Kempe, D., Kleinberg, J., and Kumar, A. (2000), Connectivity and inference problems for temporal networks, In *Proceedings of 32nd ACM Symposium on Theory of Computing*, pp. 504–513, Portland, USA: ACM.

Kempe, D., Kleinberg, J., and Tardos, E. (2003), Maximizing the spread of influence through a social network, In *Proceedings of 9th ACM International Conference on Knowledge Discovery and Data Mining*, Washington, USA: ACM.

Keränen, A. and Ott, J. (2009), DTN over aerial carriers, In *Proceedings of 4th ACM Workshop on Challenged Networks*, pp. 67–76, Beijing, China: ACM.

Kirousis, L., Kranakis, E., Krizanc, D., and Pelc, A. (2000), Power consumption in packet radio networks, *Theoretical Computer Science*, 243(1-2), 289–305.

Kossinets, G., Kleinberg, J., and Watts, D. (2008), The structure of information pathways in a social communication network, In *Proceedings of 14th International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 435–443, Las Vegas, USA: ACM.

Kostakos, V. (2009), Temporal graphs, *Physica A*, 388(6), 1007–1023.

Kuhn, F., Lynch, N., and Oshman, R. (2010), Distributed computation in dynamic networks, In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pp. 513–522, Cambridge, USA: ACM.

Kuhn, F., Moses, Y., and Oshman, R. (2011), Coordinated consensus in dynamic networks, In *Proceedings of the 30th ACM Symposium on Principles of Distributed Computing*, pp. 1–10, San Jose, California: ACM.

Kuhn, F., Wattenhofer, R., and Zollinger, A. (2003), Ad-hoc networks beyond unit disk graphs, In *Proceedings of the 2003 joint workshop on foundations of mobile computing (DIALM-POMC)*, pp. 69–78, ACM.

Lebhar, E. and Lotker, Z. (2009), Unit disk graph and physical interference model: Putting pieces together, In *Proceedings of the 23rd IEEE International Symposium on Parallel & Distributed Processing (IPDPS).*, pp. 1–8, IEEE.

Leskovec, J., Chakrabarti, D., Kleinberg, J. M., Faloutsos, C., and Ghahramani, Z. (2010), Kronecker Graphs: An Approach to Modeling Networks, *Journal of Machine Learning Research*, 11, 985–1042.

Leskovec, J., Kleinberg, J., and Faloutsos, C. (2007), Graph Evolution: Densification and Shrinking Diameters, *ACM Transactions on Knowledge Discovery from Data*, 1(1).

Lindgren, A., Doria, A., and Schelén, O. (2003), Probabilistic routing in intermittently connected networks, *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3), 19–20.

Litovsky, I., Métivier, Y., and Sopena, E. (1999), *Graph Relabelling Systems and Distributed Algorithms*, *H. Ehrig, H.J. Kreowski, U. Montanari and G. Rozenberg (Eds.),* Handbook of Graph Grammars and Computing by Graph Transformation, pp. 1–53.

Liu, C. and Wu, J. (2009), Scalable Routing in Cyclic Mobile Networks, *IEEE Transactions on Parallel and Distributed Systems*, 20(9), 1325–1338.

Lotker, Z. and Peleg, D. (2010), Structure and algorithms in the SINR wireless model, *ACM SIGACT News*, 41(2), 74–84.

Lynch, N. (1989), A hundred impossibility proofs for distributed computing, In *Proceedings of the 8th annual ACM Symposium on Principles of distributed computing (PODC)*, pp. 1–28, ACM.

Mahéo, Y., Said, R., and Guidec, F. (2008), Middleware support for delay-tolerant service provision in disconnected mobile ad hoc networks, In *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pp. 1–6, Miami, USA: IEEE.

Meguerdichian, S., Koushanfar, F., Potkonjak, M., and Srivastava, M. (2001), Coverage problems in wireless ad-hoc sensor networks, In *Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM).*, Vol. 3, pp. 1380–1387, IEEE.

Métivier, Y., Saheb, N., and Zemmari, A. (2002), Randomized local elections, *Information processing letters*, 82(6), 313–320.

Métivier, Y., Saheb, N., and Zemmari, A. (2003), Analysis of a randomized rendezvous algorithm, *Information and Computation*, 184(1), 109–128.

Moscibroda, T. (2007), The worst-case capacity of wireless sensor networks, In *6th International Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 1–10, IEEE.

Moscibroda, T., Wattenhofer, R., and Zollinger, A. (2006), Topology control meets SINR: the scheduling complexity of arbitrary topologies, In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MOBIHOC)*, pp. 310–321, ACM.

Nolan, J. (2004), Bisectored unit disk graphs, *Networks*, 43(3), 141–152.

O'Dell, R. and Wattenhofer, R. (2005), Information dissemination in highly dynamic graphs, In *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pp. 104–110, Cologne, Germany: ACM.

Papadopouli, M. and Schulzrinne, H. (2000), Seven degrees of separation in mobile ad hoc networks, In *Proceedings IEEE Global Communication Conference (GLOBECOM)*, Vol. 3, pp. 1707–1711, San Francisco, USA: IEEE.

Penrose, M. (2003), Random geometric graphs, Vol. 5, Oxford University Press Oxford.

Penrose, M. (1997), The longest edge of the random minimal spanning tree, *The annals of applied probability*, 7(2), 340–361.

Penrose, M. (1999), On k-connectivity for a geometric random graph, *Random Structures & Algorithms*, 15(2), 145–164.

Ramanathan, R., Basu, P., and Krishnan, R. (2007), Towards a formalism for routing in challenged networks, In *Proceedings of 2nd ACM Workshop on Challenged Networks (CHANTS)*, pp. 3–10.

Ros, F., Ruiz, P., and Stojmenovic, I. (2012), Acknowledgment-based Broadcast Protocol for Reliable and Efficient Data Dissemination in Vehicular Ad-hoc Networks, *IEEE Transactions on Mobile Computing*, 11(1), 33–46.

Ruiz, P., Dorronsoro, B., Bouvry, P., and Tardón, L. (2012), Information dissemination in VANETs based upon a tree topology, *Ad Hoc Networks*, 10(1), 111–127.

Santoro, N., Quattrociocchi, W., Flocchini, P., Casteigts, A., and Amblard, F. (2011), Time-Varying Graphs and Social Network Analysis: Temporal Indicators and Metrics, In *3rd AISB Social Networks and Multiagent Systems Symposium (SNAMAS)*, pp. 32–38, York, England.

Scherrer, A., Borgnat, P., Fleury, E., Guillaume, J. L., and Robardet, C. (2008), Description and simulation of dynamic mobility networks, *Comput. Netw.*, 52(15), 2842–2858.

Spyropoulos, T., Psounis, K., and Raghavendra, C. (2005), Spray and wait: an efficient routing scheme for intermittently connected mobile networks, In *Proceedings of ACM Workshop on Delay-Tolerant Networking*, pp. 252–259.

Tang, J., Musolesi, M., Mascolo, C., and Latora, V. (2010), Characterising temporal distance and reachability in mobile and online social networks, *ACM Computer Communication Review*, 40(1), 118–124.

Tang, J., Scellato, S., Musolesi, M., Mascolo, C., and Latora, V. (2010), Small-world behavior in time-varying graphs, *Phys. Rev. E*, 81(5).

Von Rickenbach, P., Schmid, S., Wattenhofer, R., and Zollinger, A. (2005), A robust interference model for wireless ad-hoc networks, In *Proceedings of the 19th parallel and distributed processing symposium (IPDPS).*, pp. 1–8, IEEE.

Wan, P. and Yi, C. (2006), Coverage by randomly deployed wireless sensor networks, *IEEE/ACM Transactions on Networking*, 14(SI), 2658–2669.

Wang, S., Torgerson, J., Schoolcraft, J., and Brenman, Y. (2009), The Deep Impact Network Experiment Operations Center Monitor and Control System, In *Proceedings of 3rd IEEE international Conference on Space Mission Challenges For information Technology*, pp. 34–40, Pasadena, USA: IEEE.

Yamashita, M. and Kameda, T. (1996), Computing on anonymous networks: Part I and II, *IEEE Trans. on Par. and Distributed Systems*, 7(1), 69 – 96.

Yick, J., Mukherjee, B., and Ghosal, D. (2008), Wireless sensor network survey, *Computer Networks*, 52(12), 2292–2330.

Zhang, D., Gogi, S. A., Broyles, D. S., Çetinkaya, E. K., and Sterbenz, J. P. (2012), Modelling wireless challenges, In *Proceedings of the 18th annual international conference on Mobile computing and networking*, Mobicom '12, pp. 423–426, New York, NY, USA: ACM.

Zhang, X., Kurose, J., Levine, B., Towsley, D., and Zhang, H. (2007), Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing, In *Proceedings of 13th ACM Int. Conference on Mobile Computing and Networking*, pp. 195–206, Montreal, Canada: ACM.

Zhang, Z. (2006), Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges, *IEEE Communications Surveys & Tutorials*, 8(1), 24–37.

Zhao, W., Ammar, M., and Zegura, E. (2004), A message ferrying approach for data delivery in sparse mobile ad hoc networks, In *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pp. 187–198.

List of acronyms/abbreviations

DTN delay-tolerant network

FSM finite state machine

SINR signal-to-interference plus noise ratio

TM Turing machine

TVG time-varying graph

UAV unmanned aerial vehicle

UDG unit disk graph

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

University of Ottawa
School of Electrical Engineering and Computer Science
800 King Edward Avenue
Ottawa, Ontario
K1N 6N5

2a. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

2b. CONTROLLED GOODS

(NON-CONTROLLED GOODS)
DMC A
REVIEW: GCEC JUNE 2010

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Deterministic algorithms in dynamic networks: Formal models and metrics

4. AUTHORS (Last name, followed by initials – ranks, titles, etc. not to be used.)

Casteigts, A.; Flocchini, P.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (Total cited in document.)
April 2013	78	126

7. DESCRIPTIVE NOTES (The category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Contract Report

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Ottawa
3701 Carling Avenue, Ottawa ON K1A 0Z4, Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
15by03	W7714-115111/001/SV

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC Ottawa CR 2013-020	

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11)) is possible, a wider announcement audience may be selected.)

13. ABSTRACT (A brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The number of telecommunication networks deployed in a dynamic environment is quickly growing. This trend exists both in everyday life (e.g., smartphones, vehicles, and commercial satellites) and in a military context (e.g., dismounted soldiers or swarms of UAVs). Unfortunately, few theoretical tools to date have enabled the study of dynamic networks in a formal and rigorous way. As a result, it is hard and sometimes impossible to guarantee, mathematically, that a given algorithm will reach its objectives once deployed in real conditions. In this report, we identify a collection of recent theoretical tools whose purpose is to model, describe, and leverage dynamic networks in a formal way. These tools include a dynamic graph formalism, various computational models, and communication models for distributed networks. We extend many graph theoretical concepts towards a dynamic variant and show how these new variants impact the solution of classical distributed problems. The report also presents a hierarchy of dynamic networks based on dynamic graph properties, thereby offering a combinatorial alternative to the well-known mobility models typically used in simulations.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

dynamic networks
wireless networks
vehicular ad hoc networks
VANET
mobile ad hoc networks
MANET
time-varying graphs
algorithms
deterministic algorithms

Defence R&D Canada

Canada's leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

www.drdc-rddc.gc.ca