Time to Change: On Distributed Computing in Dynamic Networks*

N. Santoro[1]

1 **School of Computer Science, Carleton University, Ottawa, Canada**
santoro@scs.carleton.ca

Abstract

In highly dynamic networks, topological changes are not anomalies but rather integral part of their nature. Such networks are becoming quite ubiquitous. They include systems where the entities are mobile and communicate without infrastructure (e.g. vehicles, satellites, robots, or pedestrian smartphones): the topology changes as the entities move. They also include systems, such as peer-to-peer networks, where the changes are caused by entities entering and leaving the system, They even include systems where there is no physical mobility at all, such as social networks. A vast literature on these dynamic networks has been produced in many different fields, including distributed computing. The several efforts to survey the status of the research and attempts to clarify and classify models and assumptions, have so far brought more valuable bibliographic data than order and clarity. Goal of this note is to ask questions that might bring author and readers to start to clarify some important research aspects and put some order in a sometimes confusing field. The focus here is entirely on distributed computing, specifically on its deterministic aspects.

1998 ACM Subject Classification C.2.4 Distributed Systems, F.1.1 Models of Computation, F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases distributed computing, dynamic networks, time-varying graphs, mobile agents

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Computing in networked environments has been one of the core research areas and concerns of distributed computing. The structure of such environments is modeled as a simple graph, where the nodes correspond to the computational entities and the edges to existing communication links between pairs of entities.

While assuming the network structure to be static, the research soon focused on the study of topological changes in order to model faults and failures occuring in real distributed systems. Examined in the context of *fault-tolerance*, the changes were however considered a small scale phenomenon, limited and localized. The possibility of extensive changes recurring in the lifetime of the system has been object of study within the field of *self-stabilization*: incorrect computations are allowed to take place in a period of instability; it is however assumed that the instability stops, at least long enough, so that the computation can eventually produce correct results. None of these studies can deal with systems where the topology is subject to extensive changes that can occur everywhere and possibly never stop, systems where changes are not anomalies but rather integral part of their nature.

* This work was supported by NSERC (Canada) under the Discovery Grants program.

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–13
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Such systems do indeed exist and are becoming quite ubiquitous. Such are for example systems where the entities are truly mobile and can communicate without infrastructure (e.g. vehicles, satellites, robots, or pedestrian smartphones): an edge exists between two entities if they are within communication range; the topology changes (possibly dramatically) as the entities move. In these systems end-to-end connectivity does not necessarily hold, the network might be always disconnected, still communication may be available over time and space making broadcast, routing, computations feasible. These networks have been extensively studied from engineers uder the names of delay-tolerant or challanged networks. In addition to these *ad-hoc wireless mobile networks*, the same level of dynamic changes occur in systems where there is no explicit communication, such as swarms of *autonomous mobile robots*: each robot sees the positions of the robots within its visibility range, and based on these positions, it computes a destination and moves there; the topology of the visibility graph changes during the execution of protocol. It can also occur in systems where the changes are caused by entities entering and leaving the system, such as *peer-to-peer networks*. It can even occur in systems where there is no physical mobility at all, such as *social networks*.

Due to the abundance of different contexts where these highly dynamic networks arise and their importance, a vast literature in many different fields has been produced, including by distributed computing researchers, focusing on one or another aspect of these systems. Significant efforts have been made to model and formally describe the aspects under examination; not surprising, the "lexicon" is rather confusing, with the same object being given different names (e.g., “temporal distance” introduced in [23], has been subsequently called “reachability time” in [55], “information latency” in [64], and “temporal proximity” in [65]) and sometimes the same name being used to define different classes of objects (e.g., “temporal graphs" defined in [65] *vs* the later use in [74]).

There have been several efforts to survey the status of the research, attempting to clarify and classify models and assumptions and research results (e.g. the recent [19, 57, 67, 74]), in particular the monumental effort by Holme [56]. From the distributed computing viewpoint, these efforts have so far brought more valuable bibliographic data and interesting information than order and clarity. Goal of this note is to ask questions that might bring author and readers to start to clarify some important research aspects and put some order in a sometimes confusing field.

2 What and How to Represent ?

There are many popular ways to represent dynamic networks: *temporal networks*, *evolving graphs*, *multiplex networks*; as discussed below, they are actually all equivalent and equally limited by the restrictive assumptions they make.

A less constrained general mathematical formalism that describes many different types of dynamic networks is the one offered by TVG (for *time-varying graph*) introduced in [28] and described next.

2.1 TVG

TVG is a simple model that includes the other commonly used representations as instances, plus it allows to express other (possibly more complex) computational dynamic systems.

In this model, the dynamic system is described as a *time-varying graph* $\mathcal{G} = (V, E, \mathcal{T}, \psi, \rho, \zeta)$, where V is the set vertices or nodes, representing the system entities (e.g., vehicles, robots) and $E \subseteq V \times V(\times L)$ is the set of (directed or undirected) edges representing connections (e.g. communication, contact, relation, link,...) between pairs of entities; edge can be labeled, the labels in L are domain-specific (e.g., intensity of relation, type of carrier, ...) and possibly multi-valued (e.g. <*satellite link; bandwidth of 4 MHz; encryption available;...*>).

The system exists for a contiguous interval of time $\mathcal{T} \subseteq \mathbb{R}$ called *lifetime*. The system is said to be *limited* if $\mathcal{T}$ is closed, *unlimited* otherwise; in both cases, it is generally assumed that the system has a beginning, which occurs at time $t = 0$.

The dynamics of the system is specified by the *node presence* function, $\psi : V \times \mathcal{T} \to \{0, 1\}$, and the *edge presence* function, $\rho : E \times \mathcal{T} \to \{0, 1\}$, where $\psi(x, t) = 1$ (resp., $\rho(e, t) = 1$) $\iff$ node $x \in V$ (resp., edge $e \in E$) is present at time $t \in \mathcal{T}$.

Finally, the function $\zeta : E \times \mathcal{T} \to \mathbb{R} \cup \{\perp\}$, is the *latency* (or duration, delay, ...) of the connection. So, for example, $\zeta((x, y), t) = d$ may indicate that a message from x to y, if sent at time t, will arrive at time $t + d$; or that the traversal by a mobile agent of edge (x, y), if started at time t, is completed at time $t + d$. On the other hand, $\zeta((x, y), t) = \perp$ indicates that, if starting at t, there is not enough time to use the connection (eg, send a message, perform a traversal, ...).

An important concept is that of a *snapshot* of the system at time $t \in \mathcal{T}$, denoted by $G(t) = (N(t), E(t))$, where $N(t) = \{x \in N : \psi(x, t) = 1\}$ and $E(t) = \{e \in E : \rho(x, t) = 1\}$ are the nodes and edges present at time t. The *footprint* of the system is just the aggregate graph of all footprints: $G = (V, E)$. It is assumed that the footprint is connected; otherwise, the system is considered composed of separate non-interacting dynamic systems, each with a connected footprint. Observe that connectivity of the footprint G has no implication on the connectivity of any of the snapshots $G(t)$.

The TVG model can be naturally *extended* by adding any number of (possibly temporal) functions on the nodes (e.g., $f_i : V \times \mathcal{T} \to F_i$) and/or on the edges (e.g., $h_j : E \times \mathcal{T} \to H_j$) to appropriate domains, to describe specific system features (e.g., cost, weight, energy, ...).

2.2 Synchronous Systems

The model can be also *restricted* by imposing assumptions. The most common restriction is by assuming a *discrete synchronous* system: the changes in the system occur at discrete time steps (i.e., $\mathcal{T} \subseteq \mathbb{N}$), and its dynamics is fully described as a sequence of synchronous rounds.

For discrete synchronous systems, a TVG (or aspects of it) can be equivalenty and conveniently expressed in other ways.

For example, a compact representation is by listing for each edge e the set $I(e)$ of all the contiguous intervals of time when e was present. Indeed the couple $\mathcal{I} = (N, I)$, where $I = \bigcup_{e \in E} I(e)$, is a common definition of the class of discrete synchronous systems, and it is known in the literature as *temporal networks* [57]. Note that temporal neworks do not consider node presence and more importantly they have no latency; in other words they describe discrete synchronous systems with *instantaneous contacts*.

Another popular representation for discrete synchronous systems is by considering the sequence $S(\mathcal{G}) = < G(0), G(1), G(2), \ldots, G(t), \ldots >$ of all[1] the snapshots of $\mathcal{G}$. Indeed any sequence of static graphs $S = < G_0, G_1, G_2, \ldots, G_t, \ldots >$, called *evolving graph* [44], can be seen as the sequence of snapshots of a unique TVG $\mathcal{G}$ (once the latency function has been defined). The idea of representing a dynamic graph as a sequence of static graphs was first suggested in [52]; the proposal of using a sequence of graphs to model discrete synchronous systems was made by [89] in the context of social networks, and by [44] in the

[1] A more succint representation is to consider only the snapshots where a topological change occurs.

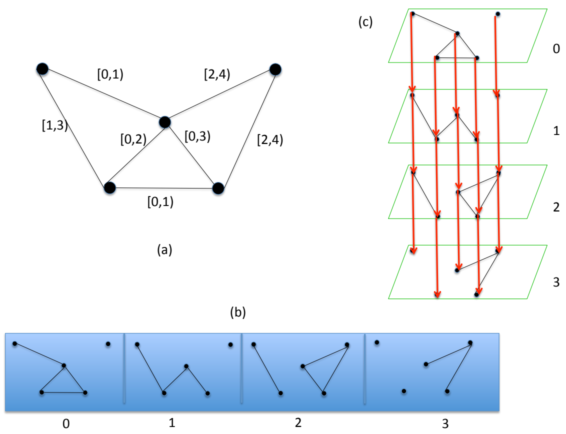


Figure 1 A discrete synchronous limited TVG represented as (a) a temporal network, (b) an evolving graph, (c) a multiplex network.

context of ad-hoc mobile networks. The evolving graph representation is perhaps the most commonly used for discrete synchronous systems (often without references and under new names). As mentioned, the latency has to be specified and added to this representation to make it equivalent to that of a (discrete synchronous) TVG.

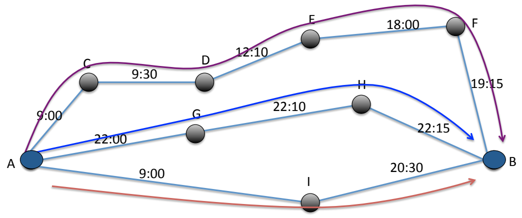
Given the sequence of snapshots $S(\mathcal{G}) = < G(0), G(1), G(2), \ldots, G(t), \ldots >$, consider the multy-layer graph $\mathcal{M}(S)$ obtained connecting $G(t)$ to $G(t+1)$ by adding an edge from each node in $G(t)$ to the same node in $G(t+1)$ (if present in both snapshots). Clearly this multi-layer graph captures the same information as the evolving graph $S(\mathcal{G})$; thus, if enanced with the specification of the latency, it becomes computationally equivalent to the TVG $\mathcal{G}$. Such a multy-layer graph $\mathcal{M}$, called *multiplex network*, is a commonly used represention of a discrete synchronous systems (for recent survey see [67]).

2.3 Journeys and Distances

A crucial concept in dynamic networks is that of *journey*, the dynamic equivalent of "walk" in static graphs. More precisely a journey is a sequence $\mathcal{J} =< (e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k) >$, where $< e_1, e_2, ..., e_k >$ is a walk in G, $\rho(e_i, t_i) = 1$ and $t_i + \zeta(e_i, t_i) \leq t_{i+1}$. That is, the walk edges are present in the graph at the appropriate times with the latency long enough so each edge can be traversed in time. Time t_1 is the start time of the journey $\mathcal{J}$, and $t_k + \zeta(e_k, t_k)$ is the time it ends, denoted by $start(\mathcal{J})$ and $end(\mathcal{J})$ respectively. Journeys could actually be infinite, in which case they have no end.

Depending on whether or not there are time gaps in the walk, we can distinguish between two types of jurneys: a journey $\mathcal{J} =< (e_1, t_1), (e_2, t_2), \ldots, (e_k, t_k) >$ is *direct* if, for all $1 \leq i < k$, $t_i + \zeta(e_i, t_i) = t_{i+1}$, i.e., there is no waiting before traversing any edge; otherwise is *indirect*. This distinction is sometimes relevant because there are dynamic networks where buffering is not supported (and thus only direct journeys are allowed). Some interesting differences between allowing and not allowing waiting have been recently established [25, 58]. In the following, we assume that waiting is allowed, and thus make no distinction between direct and undirect journeys.

A finite journey is a walk over time from a source to a destination and therefore has not



shortest	A,I,B	min # hops	(2 hops)
foremost	A,C,D,E,F	earliest arrival	(19:15)
fastest	A,G,H,B	smallest duration	(15 minutes)

Figure 2 Three types of minimal journeys; in this example latency is 0.

only a *topological* length $|\mathcal{J}|$, defined as the number of edges in the walk, but also a *temporal* length $||\mathcal{J}|| = end(\mathcal{J}) - start(\mathcal{J})$ defined as the time elapsed to perform the walk.

This gives rise to distinct definitions of *distance* in a time-varying graph $\mathcal{G}$:
`shortest distance`: $d(u, v, t) = Min\{|\mathcal{J}| : \mathcal{J} \in \mathcal{J}_{(u,v)} \wedge start(\mathcal{J}) \geq t\}$ (i.e., min hop);
`fastest distance`: $\delta(u, v, t) = Min\{||\mathcal{J}|| : \mathcal{J} \in \mathcal{J}_{(u,v)} \wedge start(\mathcal{J}) \geq t\}$ (i.e., min duration);
`foremost distance`: $\partial(u, v, t) = Min\{end(\mathcal{J}) : \mathcal{J} \in \mathcal{J}_{(u,v)} \wedge start(\mathcal{J}) \geq t\}$ (i.e., ends first);
where $\mathcal{J}_{(u,v)}$ denotes the set of all journeys from u to v.

Indeed, for all all the classical measures of static graphs and networks (diameter, degree, eccentricity, centrality, etc.) there exist (one or more) temporal counterparts (temporal diameter, temporal degree, temporal eccentricity, temporal centrality, etc.).

3 What to Investigate ?

3.1 Dead or Live, Centralized or Distributed ?

Most of the existing algorithmic investigations and results assume global a-priori knowledge of the system; that is, the entire graph $\mathcal{G}$ is given as an input to the computation. In other words, they consider dynamic networks that are (not only discrete synchronous but also) *limited*; the investigation is *post-mortem*, i.e. after the system has ended its limited life-time (and no more data is being produced); and the computation is off-line, totally *centralized*. That is, they are *centralized investigations of dead systems*. This is for example the case of [16, 23, 44, 48, 53, 59, 64, 78, 73], and in particular of all the investigations on data previously collected from real dynamic networks (e.g., [62, 63, 66, 82, 84, 85, 92]).

The interest of this note is however on *distributed computations in dynamic networks*. This means that the computation is distributively performed *inside* the time-varying graph. In other words, the system is *live*; its lifetime $\mathcal{T}$ is *unlimited* (as far as the computation is concerned); and the computation is *decentralized* and *localized*.

The lifetime $\mathcal{T}$ of $\mathcal{G}$ is divided in three parts: the instantaneous *present*, $\hat{t}$; the limited *past*, $past(\hat{t}) = \{t \in \mathcal{T} : 0 < t < \hat{t}\}$; and the unlimited *future*, $future(\hat{t}) = \{t \in \mathcal{T} : t > \hat{t}\}$. Each computational entity (node, web site, vehicle, ...) operates in the present, and is aware only of the *local events*, i.e., topological changes in which it is involved; it might remember its past (if it has enough memory); however, it does not know the future.

3.2 Who is in Control ?

To understand how to deal with the future, it is important to understand the relationship between the changes occuring in the system and the computation performed by the entities. To ask what is causing the system changes is important but it does not necessarily clarify the nature of the relationship. For example, in ad-hoc wireless mobile networks, it is the movement of the entities that causes the topological changes; similarly, in robotic swarms or in mobile sensor networks, the changes are generated by the movement of the entities. Apparently, in all these systems, the cause of changes is the same: the entities' movements; there is however a fundamental difference between the former and the latter systems.

In the latter, the movement of an entity is influenced by the computation: where an entity goes next (and thus what new edges are being formed) is determined by the protocol; in other words, the *computation generates the graph*. This means that we (algorithm designers) could program the entities so to construct graphs with specific properties. This indeed is what happens when we design protocols that allow autonomous mobile robots with limited visibility to arrange themselves in space so to form a specific geometric pattern, or mobile sensors to spread over a territory so to homogenously cover it. We shall call these types of situations as of *controlled generation* of the graph.

Totally different is the first type of systems: the movements are independent of the computation (e.g., broadcast, routing, etc) performed by the entities. More precisely, the computation has no control over the topological changes of the system. We call this type of situation as of an *uncontrolled generation* of the graph; in this situation, we actually envision the changes as caused by an adversary operating against the computation.

It is on this type of systems that we focus in the rest of this note.

3.3 What Problems ?

In the live systems we consider, the computational entities operate in a decentralized and localized matter in an ever changing scenario, without control over the changes. The research investigations on these systems have been both intensive and extensive, carried out mainly within the engineering community.

The distributed computing focus has been on a variety of classical problems, such as *information propagation*: routing, multicast, broadcast, gossip, etc. (e.g. [4, 5, 13, 14, 26, 27, 31, 32, 33, 35, 42, 54, 83, 93]); *coordination*: aggregation, naming, counting, etc. [3, 20, 34, 40, 76, 79]; *computability* (e.g., [7, 8, 21, 24, 28, 36, 39, 69, 75]); *control*: election, consensus, synchronization, etc. (e.g., [6, 10, 11, 15, 17, 18, 30, 36, 49, 50, 61, 68, 70, 88].

A separate area of research has been on computations by entities opportunistically moving from node to node by traversing edges when they appear. This is the classical environment of mobile agents (or robots) moving in a network, extended to dynamic networks. Also this case is one of uncontrolled generation, as the mobile agents have no control over the changes in the network; the changes are often seen as generated by the movement of *carriers*. The main research focus has been on *search* and *exploration* (e.g., [1, 2, 12, 22, 37, 41, 43, 45, 46, 58]).

4 Without Control What to Assume ?

The fact that the future is unknown and under the control of an adversary means that, in order to be able to perform some meaningful computation, some *assumptions* have to be made. Usually called *a priori knowledge* and sometimes *oracles*, these assumptions restrict the universe under observation, limiting the power of the adversary.

As mentioned before, the most common assumption is that the system is *discrete synchronous.* Another common (usually hidden) assumption is that the footprint $G = (V, N)$ is finite, i.e., both N and E are finite. Let us make these assumptions. Still, they are not enough, and additional assumptions are necessary.

For example, in the non-deterministic realm, additional assumptions are made on the probability of the appearance of every edge (e.g, it obeys a Poisson process) or on the relationship beween successive snapshots $G(t)$ and $G(t+1)$ (e.g., edge-Markovian process); e.g., see [14, 29, 31, 33]. The focus of this note is however on the *deterministic* side.

4.1 Frequency Assumptions

A type of deterministic additional assumptions are about the *frequency* of the changes. Let an edge $e \in E$ be called *transient* if it appears in a finite number of snapshots $G(t)$, *recurrent* otherwise. Notice that if there are both transient and recurrent edges, there exists a time $\tilde{t}$ after which all appearing edges are recurrent.

We say that the system $\mathcal{G}$ is *recurrent* if all edges are recurrent: $\forall e \in E, t \in, \exists t' > t : e \in E(t')$; this restriction is sometimes called *local fairness.* Example of recurrent systems are population protocols with a fair scheduler. Investigations include e.g., [1, 2, 6, 7, 8, 26, 27, 75].

A more restricted class is that of B*-bounded* recurrent systems, $B \in \mathbb{N}$, defined by the assumption $e \in E(t) \Rightarrow e \in E(t')$ where $t < t' \leq t + B$ (e.g., [1, 2, 26, 27]).

Even more restricted is the class of P*-periodic* systems, $P \in \mathbb{N}$, defined by the assumption $e \in E(t) \Rightarrow e \in E(t+P)$. Examples of periodic systems are public transports with fixed timetable, low-earth orbiting satellite (LEO) systems. Study of computing in periodic systems include [1, 2, 26, 27, 45, 46, 58, 60, 71, 72]. Notice that to determine whether or not a system is periodic is undecidable. Similiarly undecidable is to verify if a periodic system has period P. In other words, periodicity without knowledge of the period is not a useful assumption.

Note that all these frequency assumptions are restictions on the functions ψ and ρ. Another type of requirement sometimes imposed on those functions is that they should somehow reflect the behaviour of "real life" systems. With this motivation, the engineering community has developed and has been using several *mobility patterns*, i.e. restrictions on the functions ψ and ρ to mimic experimentally observed changes due to mobility of vehicles, humans, etc. (e.g., [38, 81, 87]).

4.2 Connectivity Assumptions

Of all the common assumptions we considered so far, none has any impact on the *connectivity* of the network. Indeed simultaneous end-to-end connectivity might not be guaranteed, and it is also possible that all snapshots $G(t)$ might be disconnected in spite of those assumptions. Not surprising, especially for discrete synchronous systems, a popular class of additional assumptions are those relating to *connectivity.*

The weakest such assumption is *temporal connectivity*, that is $\forall x, y \in V, t \in \mathcal{T}$ there exixts a journey $J \in \mathcal{J}_{(u,v)}$ such that $start(J) \geq t$. This assumption is typical in the engineering investigations; it is also the one used in the pioneering work of Awerbuch and Even [13].

Stronger assumptions require simultaneous end-to-end connectivity to hold at some point in time. In increasing order of requirement's strenght, we have *recurrent connectivity*: $\forall t \in \mathcal{T}, \exists\ t' \geq t :\ G(t')$ is connected (e.g., [10, 61]); B*-bounded connectivity* : $\forall t \in \mathcal{T}, \exists\ t' \leq t + B : G(t')$ is connected; and P*-periodic connectivity* : $\forall t \in \mathcal{T}, \exists\ t' \geq t : \forall j \in G(t' + jP)$ is connected.

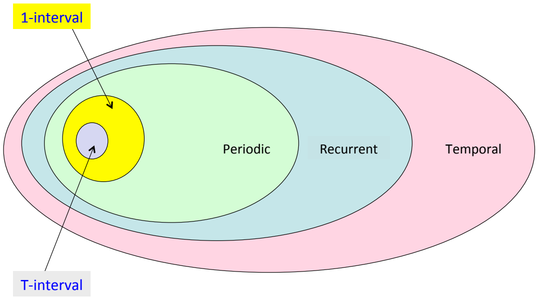


Figure 3 Connectivity assumptions.

Finally, *permanent connectivity*: $\forall t \in \mathcal{T}, G(t)$ is connected; this assumption is also known as *1-interval connectivity*. An even more stringent assumption is *permanent connectivity with persistent backbone* which requires that the same connected spanning subgraph persists for $T > 1$ consecutive snapshots: $\forall t \in \mathcal{T}, G_T(t) = \bigcap_{0 \le j < T} G(t+j)$ is connected; this assumption is also known as *T-interval connectivity*. Both permanent and T-interval connectivity are often assumed in distributed computations (e.g., [3, 41, 43, 54, 59, 68, 69, 70, 83, 88]).

4.3 Power of Assumptions

Notice that to each set of assumptions corresponds the class of systems satisfying those assumptions. Important questions are about the *computational power* of these classes of systems. For example, is one class more powerful than another ? What is the weakest class where a given problem is solvable ? (i.e., what are the weakest assumptions which allow to solve a given problem ?)

For example, with respect to the frequency of changes, we have identified the classes $\mathcal{G}[Recurrent]$, $\mathcal{G}[Bounded]$, and $\mathcal{G}[Periodic]$. Let $\mathcal{P}[Recurrent]$, $\mathcal{P}[Bounded]$, and $\mathcal{P}[Periodic]$ be the set of problems solvable in those classes. Clearly, $\mathcal{P}[Recurrent] \subseteq \mathcal{P}[Bounded] \subseteq \mathcal{P}[Periodic]$. Consider the problem of *minimal broadcast with termination detection*: optimally diffusing some information and the source knowing (within finite time) of the completion of the process. As we discussed previously, with respect to "minimality", there are three types of journeys and thus of broadcasts: *foremost* broadcast *FoB*, in which the date of delivery is minimized at every node; *shortest* broadcast *ShB*, where the number of hops used by the broadcast is minimized relative to every node; and *fastest* broadcast *FaB*, where the overall duration of the broadcast is minimized (however late the departure be). Interestingly : $FoB \in \mathcal{P}[Recurrent]$ but $ShB \notin \mathcal{P}[Recurrent]$; furthermore $ShB \in \mathcal{P}[Bounded]$ but $FaB \notin \mathcal{P}[Bounded]$; on the other hand, $FaB \in \mathcal{P}[Periodic]$. This implies first of all a strict order on the difficulty of the three problems:

$$FoB < ShB < FaB$$

It also shows that the inclusion among the set of problems is strict:

$$\mathcal{P}[Recurrent] \subsetneq \mathcal{P}[Bounded] \subsetneq \mathcal{P}[Periodic]$$

and thus the strict hierarchy of computational power of those graph classes:

$$\mathcal{G}[Recurrent] \prec \mathcal{G}[Bounded] \prec \mathcal{G}[Periodic]$$

Note that these results, established in [26, 27], hold even without the discrete synchronous assumption.

It is clear that any result established under a set of assumptions gives some information about the computational impact of those assumptions. This information gives raise to a hierarchy of classes of systems, as pointed out in [28].

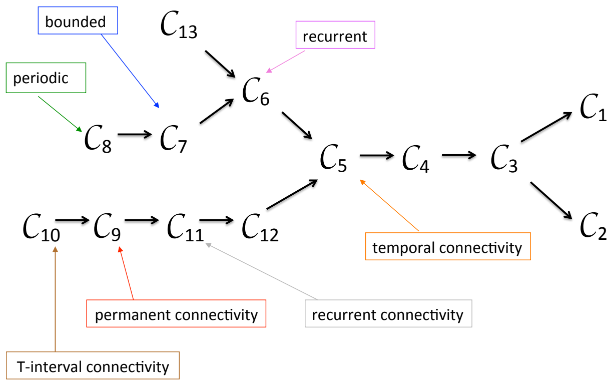


Figure 4 Hierarchy of system classes, from [28].

5 Conclusions

At the end of this note, it should be clear that researching "distributed computing in dynamic networks" means to consider a system that is (still) alive and evolving, and to understand that the computation is by necessity decentralized and localized. More precisely, the point of view is that of the computational entity, operating in an ever changing system; always in the present, the entity is only aware of the changes in which it is involved, possibly remembering the past, and in general without knowledge of the future.

The first distinction is on whether or not the computational entity has control over the topological changes in which it is involved; that is, on whether the generation is controlled or uncontrolled by the computation. In the case of controlled generation, an entity has a degree of control over the future. In the case of uncontrolled generation (a condition that occurs also in the case of mobile agents), assumptions have to be made to render computation possible. The next distinction is on what assumptions are made.

Among the open research goals the obvious ones are to establish new results and to explore new problems. An important goal is to *remove assumptions*. Indeed, when facing a problem the guiding question should be: what is the weakest assumption that makes the problem solvable?

The strongest assumption commonly made is that the system is discrete synchronous. Actually almost all the algorithmic investigations making this assumption further assume that the scheduling of the entities is *fully sysnchronous*: at each time steps, all the entities are active and participate in the distributed computation. A weaker assumption is that of a *semi-synchronous* setting: at each time step a (non-empty) subset of the entities are active, the activation choice made by a fair but adversarial scheduler. This model, quite common in the context of robotic swarms [47], has just started to be examined in the other dynamic networks contexts [41].

A larger open research direction is to remove the discrete synchronous assumption completely, and look at the more general case possible, asynchronous and continuous. As mentioned, some investigations have been carried out and results established in the general case (e.g., [26, 27, 29]). Interestingly, in some investigations that assume discrete synchronous systems, the analysis is however carried out in the continuous setting (e.g., [65]).

References

1 E. Aaron, D. Krizanc, and E. Meyerson. DMVP: Foremost waypoint coverage of time-varying graphs. In *Proc. 40th Int. Work. Graph Th. Conc. Comp. Sci. (WG)*, 29–41, 2014.

2 E. Aaron, D. Krizanc, and E. Meyerson. Multi-robot foremost coverage of time-varying graphs In *Proc. 10th ALGOSENSORS*, 22-38, 2015.

3 S. Abshoff and F. Meyer auf der Heide. Continuous aggregation in dynamic ad-hoc networks. In *Proc. 21st Int. Coll. on Structural Inf. and Comm. Compl. (SIROCCO)*, 194-209, 2014.

4 S. Abshoff, M. Benter, M. Malatyali, and F. Meyer auf der Heide. On two-party communication through dynamic networks In *Proc. 17th International Conference on Principles of Distr. Syst. (OPODIS)*, 11–22, 2013.

5 M. Ahmadi, A. Ghodselahi, F. Kuhn, and A.R. Molla. The cost of global broadcast in dynamic radio networks. In *these Proceedings (OPODIS)*, 2015.

6 D. Alistarh and R. Gelashvili. Polylogarithmic-time leader election in population protocols. In *Proc. 42nd Int. Coll. Automata, Languages, Program. (ICALP)*, 479–491, 2015.

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

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

9 M. Antony and A. Gupta. Finding a small set of high degree nodes in time-varying graphs. In *Proc. 15th IEEE Int. Symp. Wireless, Mob. Multimedia Netw. (WoWMoM)*, 1–6, 2014.

10 L. Arantes, F. Greve, P. Sens, and V. Simon. Eventual leader election in evolving mobile networks. In *Proc. of 17th Int. Conf. Principles of Distr. Syst. (OPODIS)*, 23–37, 2013.

11 J. Augustine, G. Pandurangan, and P. Robinson. Fast Byzantine agreement in dynamic networks. In *Proc. of 32nd Symp. Principles of Dist. Comp. (PODC)*, 74-83, 2013.

12 C. Avin, M. Koucky, and Z. Lotker. How to explore a fast-changing world. In *Proc. of the 35th Int. Coll. on Automata, Languages and Programming (ICALP)*, 121–132, 2008.

13 B. Awerbuch and S. Even. Efficient and reliable broadcast is achievable in an eventually connected network. In *Proc. of 3rd Symp. Princip. Dist. Comp.(PODC)*, 278–281, 1984.

14 H. Baumann, P. Crescenzi, and P. Fraigniaud. Parsimonious flooding in dynamic graphs. In *Proc. of the 28th ACM Symp. on Principles of Distr. Comp. (PODC)*, 260–269, 2009.

15 A. Benchi, P. Launay, and F. Guidec. Solving consensus in opportunistic networks. In *Proc. 16th Int. Conf. on Distributed Computing and Networking (ICDCN)*, 1:1–1:10, 2015.

16 S. Bhadra and A. Ferreira. Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks. In *Proc. 2nd Intl. Conf. on Ad Hoc Networks and Wireless (ADHOC-NOW)*, 259–270, 2003.

17 M. Biely, P. Robinson, and U. Schmid. Agreement in directed dynamic networks. In *Proc. of the 19th Int. Coll. on Structural Inf. and Comm. Complexity (SIROCCO)*, 73–84, 2012.

18 M. Biely, P. Robinson, U. Schmid, M. Schwarz, and K. Winkler. Gracefully degrading consensus and k-set agreement in directed dynamic networks. In *Proc. of the 2nd International Conference on Networked Systems*, 2015.

19 B. Blonder, T.W. Wey, A. Dornhaus, R. James, and A. Sih. Temporal dynamics and network analysis. *Methods in Ecology and Evolution* (3): 958–972, 2012.

20 Q. Bramas and S. Tixeuil. The complexity of data aggregation in static and dynamic wireless sensor networks. In *Proc. 17th Int. Symp. Stab. Safety Secur. (SSS)*, 36-50, 2015.

21 P. Brandes and F. Meyer auf der Heide. Distributed computing in fault-prone dynamic networks. In *Proc. of 4th Int. Work. Theor. Aspects Dynamic Distr. Syst.*, 9–14, 2012.

22 B. Brejova, S. Dobrev, R. Kralovic, and T. Vinar. Efficient routing in carrier-based mobile networks. *Theoretical Computer Science*, 509:113–121, 2013.

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

24 A. Casteigts, S. Chaumette, and A. Ferreira. Characterizing topological assumptions of distributed algorithms in dynamic networks. In *Proc. of 16th Int. Coll. on Structural Information and Communication Complexity (SIROCCO)*, 126–140, 2009.

25 A. Casteigts, P. Flocchini, E. Godard, N. Santoro, M. Yamashita On the expressivity of time-varying graphs. *Theoretical Computer Science*, 590: 27-37, 2015

26 A. Casteigts, P. Flocchini, B. Mans, and N. Santoro. Measuring temporal lags in delay-tolerant networks. *IEEE Trans. Comp.* 63(2): 397–410, 2014.

27 A. Casteigts, P. Flocchini, B. Mans, and N. Santoro. Shortest, fastest, and foremost broadcast in dynamic networks. *Int. J. of Foundations of Comput. Sci.*, 26(4): 499–522, 2015.

28 A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. *Int. J. Parallel, Emergent and Distributed Syst.*, 27(5):387–408, 2012.

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

30 B. Charron-Bost, M. Fugger, and T. Nowak. Approximate consensus in highly dynamic networks: The role of averaging algorithms. In *Proc. 42nd International Colloquium on Automata, Languages, and Programming (ICALP)*, 528-539, 2015.

31 A.E.F. Clementi, C. Macci, A. Monti, F. Pasquale, and R. Silvestri. Flooding time of edge-Markovian evolving graphs. *SIAM J. on Discrete Mathematics*, 24(4):1694–1712, 2010.

32 A.E.F. Clementi, A. Monti, F. Pasquale, and R. Silvestri. Information spreading in stationary Markovian evolving graphs. *IEEE Trans. Par. Distr. Syst.*, 22(9):1425–1432, 2011.

33 A.E.F. Clementi, R. Silvestri, and L. Trevisan. Information spreading in dynamic graphs. *Distributed Computing* 28(1): 55-73, 2015.

34 A. Cornejo, S. Gilbert, and C. Newport. Aggregation in dynamic networks. In *Proc. Symp. Principles of Distributed Computing (PODC)*, 195-204, 2012.

35 A. Cornejo, C. Newport, S. Gollakota, J. Rao, and T.J. Giuli. Prioritized gossip in vehicular networks. *Ad Hoc Networks*, 11(1):397–409, 2013.

36 E. Coulouma and E. Godard. A characterization of dynamic networks where consensus is solvable. In *Proc. 20th Int. Coll. Structural Inf. Comm. Comp. (SIROCCO)*, 24–35, 2013.

37 O. Denysyuk and L. Rodrigues. Random walks on evolving graphs with recurring topologies. In *Proc. 28th Int. Symp. on Distributed Computing (DISC)*, 333–345, 2014.

38 A. Diab and A. Mitschele-Thiel. *Human Mobility Patterns.* IGI 2014.

39 G.A. Di Luna and R. Baldoni Non trivial computations in anonymous dynamic networks. In *these Proceedings (OPODIS)*, 2015.

40 G.A. Di Luna, R. Baldoni, S. Bonomi, and I. Chatzigiannakis. Conscious and unconscious counting on anonymous dynamic networks. In *Proc. 15th Int. Conf. on Dist. Comp. and Networking (ICDCN)*, 257–271, 2014.

41 G.A. Di Luna, S. Dobrev, P. Flocchini, and N. Santoro. Live exploration of dynamic rings *arXiv*: 1512.05306, 2015.

42 C.Dutta, G.Pandurangan, R.Rajaraman, Z.Sun, and E.Viola. On the complexity of information spreading in dynamic networks. In *Proc. Symp. on Disc. Alg. (SODA)* 717–736, 2013.

43 T. Erlebach, M. Hoffmann, and F. Kammer. On temporal graph exploration In *Proc. of 42nd Int. Coll. on Automata, Languages, and Programming (ICALP)*, 444-455, 2015.

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

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

46 P. Flocchini, B. Mans, and N. Santoro. On the exploration of time-varying networks. *Theoretical Computer Science*, 469: 53–68, 2013.

47 P. Flocchini, G. Prencipe, and N. Santoro. *Distributed Computing by Oblivious Mobile Robots.* Morgan & Claypool, 2012.

48 E. Godard and D. Mazauric. Computing the dynamic diameter of non-deterministic dynamic networks is hard. In *Proc. 10th ALGSENSORS*, 88-102, 2014.

49 C. Gomez-Calzado, A. Lafuente, M. Larrea, and M. Raynal. Fault-tolerant leader election in mobile dynamic distributed systems. In *Proc. 19th Pacific Rim Int. Symp. on Depend. Comput. (PRDC)*, 78–87, 2013.

50 F. Greve, P. Sens, L. Arantes, and V. Simon. Eventually strong failure detector with unknown membership. *The Computer Journal*, 55(12): 1507–1524, 2012.

51 B. Haeupler and F. Kuhn. Lower bounds on information dissemination in dynamic networks. In *Proc. of 26th Int. Symp. on Distributed Computing (DISC)*,166-180, 2012.

52 F. Harary and G. Gupta. Dynamic graph models. *Math. Comp. Model.*, 25(7):79–88, 1997.

53 D. Ilcinkas, R. Klasing, and A.M. Wade. Exploration of constantly connected dynamic graphs based on cactuses. In *Proc. 21st SIROCCO*, 250–262, 2014.

54 B. Haeupler, F. Kuhn. Lower bounds on information dissemination in dynamic networks. In *Proc. 26th Int. Symp.on Distributed Computing (DISC)*, 166-180, 2012.

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

56 P. Holme. Modern temporal network theory: A colloquium. *Eur. Phys. J. B*, 88: 234, 2015.

57 P. Holme and J. Saramaki. Temporal networks. *Physics Reports*, 519: 97-125, 2012.

58 D. Ilcinkas and A.M. Wade. On the power of waiting when exploring public transportation systems. *Proc. 15th Int. Conf. on Principles of Dist. Syst. (OPODIS)*, 451–464, 2011.

59 D. Ilcinkas and A.M. Wade. Exploration of the T-Interval-connected dynamic graphs: the case of the ring. In *Proceedings 20th SIROCCO*, 13-23, 2013.

60 R. Jathar, V. Yadav, and A. Gupta. Using periodic contacts for efficient routing in delay tolerant networks. *Ad Hoc & Sensor Wireless Networks*, 22(1,2): 283-308, 2014.

61 D. Jeanneau, T. Rieutord, L. Arantes, and P. Sens A failure detector for k-set agreement in asynchronous dynamic systems. INRIA Research Report 8727, 2015.

62 H. Kim, R. Anderson. Temporal node centrality in complex networks. *Phys. Rev. E*, 85: 1–8, 2012

63 M. Konschake, H.H.K. Lentz, F.J. Conraths, P. Hovel, and T. Selhorst. On the robustness of in-and out-components in a temporal network. *PloS One*, 8(2): e55223, 2013.

64 G. Kossinets, J. Kleinberg, and D. Watts. The structure of information pathways in a social communication network. In *Proceedings of the 14th Int. Conference on Knowledge Discovery and Data Mining (KDD)*, 435–443, 2008.

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

66 L. Kovanen, M. Karsai, K. Kaski, J. Kertesz, J. Saramaki. Temporal motifs in time-dependent networks. *J. Stat. Mech. Theor. Exp.*, 2011(11): 11005, 2011.

67 M. Krivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *J. Complex Networks*, 2(3): 203–271, 2014.

68 F. Kuhn, T. Locher, and R. Oshman. Gradient clock synchronization in dynamic networks *Theory of Computing Systems*, 49(4): 781 - 816, 2011.

69 F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proc. of the 42nd ACM Symp. on Theory of Computing (STOC)*, 513–522, 2010.

70 F. Kuhn, Y. Moses, and R. Oshman. Coordinated consensus in dynamic networks. In *Proc. 30th Symp. on Principles of Distributed Computing (PODC)*, 1–10, 2011.

71 C. Liu and J. Wu. Scalable routing in cyclic mobile networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(9): 1325–1338, 2009.

72 C. Mergenci and I. Korpeoglu. Routing in delay tolerant networks with periodic connections. *EURASIP J. Wireless Communications and Networking*, 202, 2015.

73 G. B. Mertzios, O. Michail, I. Chatzigiannakis, and P. G. Spirakis. Temporal network optimization subject to connectivity constraints. In *Proc. 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, 657-668, 2013.

74 O. Michail. An introduction to temporal graphs: An algorithmic perspective. In *Algorithms, Probability, Networks, and Games*, Springer, 308-343, 2015.

75 O. Michail, I. Chatzigiannakis, and P.G.. Spirakis. Mediated population protocols. *Theor. Comput. Sci.*, 412(22): 2434–2450, 2011.

76 O.Michail, I.Chatzigiannakis, and P.G.Spirakis. Naming and counting in anonymous unknown dynamic networks In *Proc. 15th Int. Symp. Stab., Safety, Sec. (SSS)*, 281-295, 2013.

77 O. Michail, I. Chatzigiannakis, and P.G. Spirakis. Causality, influence, and computation in possibly disconnected synchronous dynamic networks. *Journal of Parallel and Distributed Computing*, 74(1): 2016-2026, 2014.

78 O. Michail and P.G. Spirakis. Traveling salesman problems in temporal graphs. In *Proc. 39th Int. Symp. on Mathematical Foundations of Computer Science (MFCS)*, 553-564, 2014.

79 A. Milani and M.A. Mosteiro. A faster counting algorithm for anonymous dynamic networks. In *these Proceedings (OPODIS)*, 2015.

80 A. Mtibaa, A. Chaintreau, L. Massoulie, and C.Diot. The diameter of opportunistic mobile networks. In *Proc 3rd Int. Conf. Emerging Networking Exp. Technol. (CoNEXT)*, 12, 2007

81 M. Musolesi and C. Mascolo. A community based mobility model for ad hoc network research. In *Proc 2nd Int. Work. Multi-Hop Ad Hoc Networks*, 31–38, 2006.

82 V. Nicosia, J. Tang, M. Musolesi, G. Russo, C. Mascolo, V. Latora. Components in time-varying graphs. *Chaos*, 22(2): 023101, 2012.

83 R. O'Dell and R. Wattenhofer. Information dissemination in highly dynamic graphs. In *Proc. of the Joint Workshop on Foundations of Mobile Computing*, 104–110, 2005.

84 Y. Pan and X. Li. Structural controllability and controlling centrality of temporal networks. *PLOS ONE*, 9(4): e94998, 2014.

85 U. Redmond, M. Harrigan, and P. Cunningham. Identifying time-respecting subgraphs in temporal networks. In *Proceedings Europ. Conf. on Machine Learning*. 2012

86 F.J. Ros and P.M. Ruiz. Minimum broadcasting structure for optimal data dissemination in vehicular networks. *IEEE Transactions on Vehicular Technology*, 62(8):3964–3973, 2013.

87 A.K. Saha and D. Johnson. Modeling mobility for vehicular ad-hoc networks. In *Proceedings 1st ACM Int. Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004

88 G. Sharma and C. Bush. Distributed queueing in dynamic networks. *Parallel Processing Letters*, 25(2), 2015.

89 T.A.B. Snijders. The statistical evaluation of social network dynamics. In *Sociological Methodology* (M.E. Sobel and M.P. Becker Eds), Blackwell, 361-395, 2001.

90 J. Tang, S. Scellato, M. Musolesi, C. Mascolo, and V. Latora. Small-world behavior in time-varying graphs. *Physical Review E*, 81(5):055101, 2010.

91 J. Whitbeck, M. Dias de Amorim, V. Conan, and J.-L. Guillaume. Temporal reachability graphs. In *Proceedings 8th International Conference on Mobile Computing and Networking (MOBICOM)*, 377–388, 2012.

92 M. Wildemann, M. Rudolf, and M. Paradies. The time has come: Traversal and reachability in time-varying graphs. In *Proc. VLDB Workshop on Big-Graphs Online Querying*, 2015.

93 Z. Yang, S. Yat-sen, W. Wu, Y. Chen, and J. Zhang. Efficient information dissemination in dynamic networks. In *Proc. 42nd Int. Conf. on Parallel Proces. (ICPP)*, 603–610, 2013.

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