

Self-stabilizing Systems in Spite of High Dynamics

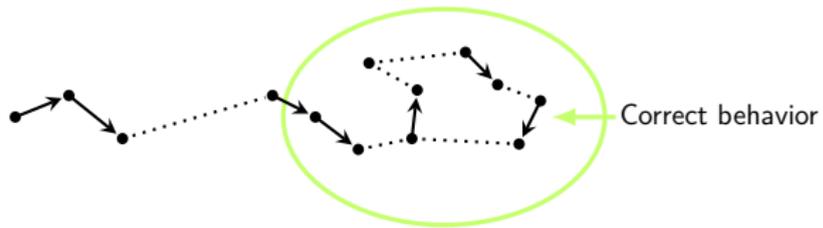
Karine Altisen¹ Stéphane Devismes¹ Anaïs Durand^{2,4}
Colette Johnen³ Franck Petit⁴

¹ VERIMAG, Grenoble
² LIMOS, Clermont Ferrand
³ LaBRI, Bordeaux
⁴ LIP6, Paris

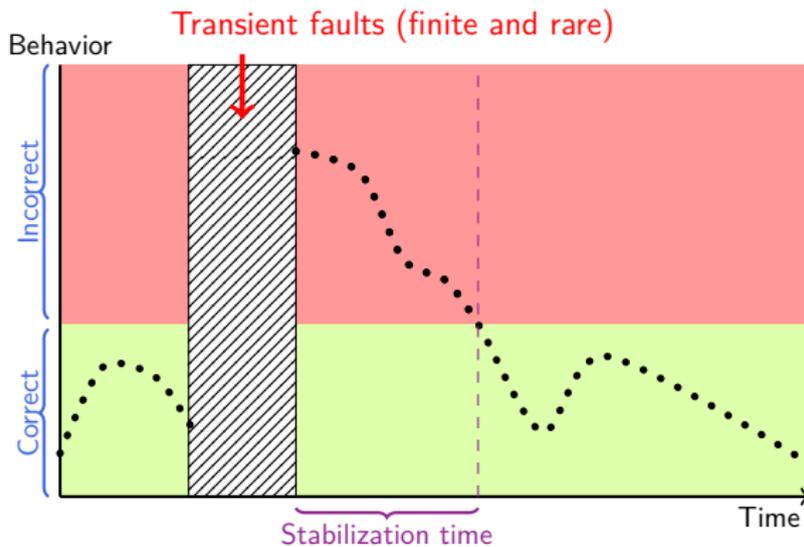
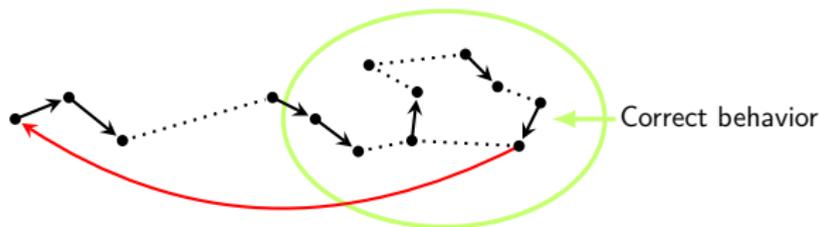
Meeting DESCARTES, October 10 2019, Paris



Self-stabilization, [Dijkstra, ACM Com., 74]



Self-stabilization, [Dijkstra, ACM Com., 74]



General approach for tolerating transient faults

Some results:

Process Crashes: FTSS [Beauquier and Kekkonen-Moneta, PODC'97]

Byzantine failures strict stabilization [Nesterenko and Arora, ICDCS'02]

Intermittent Faults intermittent lost, duplication, or reordering of messages [Delaët and Tixeuil, JPDC, 2002]

Some results:

Process Crashes: FTSS [Beauquier and Kekkonen-Moneta, PODC'97]

Byzantine failures strict stabilization [Nesterenko and Arora, ICDCS'02]

Intermittent Faults intermittent lost, duplication, or reordering of messages [Delaët and Tixeuil, JPDC, 2002]

Even with an high failure rate, convergence is possible

Some results:

Process Crashes: FTSS [Beauquier and Kekkonen-Moneta, PODC'97]

Byzantine failures strict stabilization [Nesterenko and Arora, ICDCS'02]

Intermittent Faults intermittent lost, duplication, or reordering of messages [Delaët and Tixeuil, JPDC, 2002]

Even with an high failure rate, convergence is possible

However: **static communication networks**

Self-stabilization in presence of Topological Changes

If topological are **locally detected** and **infrequent**, then a self-stabilizing algorithm designed for arbitrary topologies is well-suited.

Moreover:

Superstabilization: quick convergence after few topological changes from a legitimate configuration [Dolev and Herman, Chicago Journal of Theoretical Computer Science, 1997]

Gradual Stabilization: a generalization of superstabilization [Altisen et al., JPDC, 2019]

However: **topological changes should be transient**

Self-stabilization in Highly Dynamic Systems

Few results

Negative result: Even if the network is **always connected over the time**, **silent self-stabilization is impossible!**
[Braud-Santoni *et al.*, IJNC, 2016]

Silence: converges within finite time to a configuration from which the values of the communication registers used by the algorithm remain fixed.

Self-stabilization in Highly Dynamic Systems

Few results

Negative result: Even if the network is **always connected over the time**, **silent self-stabilization is impossible!**
[Braud-Santoni *et al.*, IJNC, 2016]

Silence: converges within finite time to a configuration from which the values of the communication registers used by the algorithm remain fixed.

Positive result: Self-stabilizing exploration of a highly dynamic **ring** by a cohort of **synchronous robots** [Bournat *et al.*, TCS, 2019]

Robot: visibility sensors, moving actuators, yet no communication capabilities.

However, only one edge maybe missing at a time
(the network is **always connected over the time**)

- Self-stabilization in **highly dynamic message-passing** systems
- Dynamics modeled as **Time-Varying Graphs (TVG)**
[Casteigts *et al.*, IJPEDES, 2012]

- Self-stabilization in **highly dynamic message-passing** systems
- Dynamics modeled as **Time-Varying Graphs (TVG)**
[Casteigts *et al.*, IJPEDES, 2012]

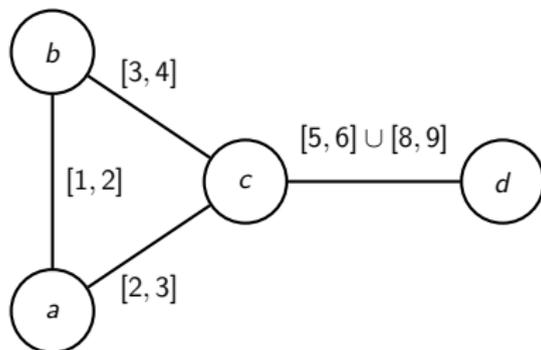
We look for (non-silent) self-stabilizing algorithm for
general classes of TVGs

(*e.g.*, we do not enforce the network to be in a particular topology at a given time)

Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function



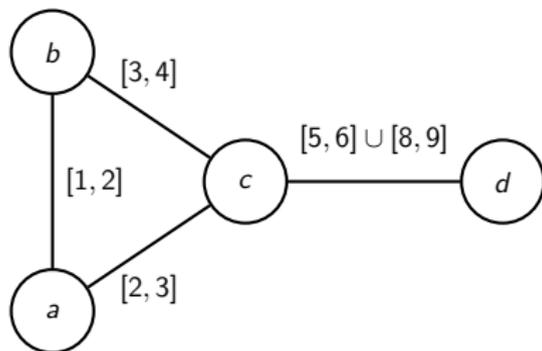
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



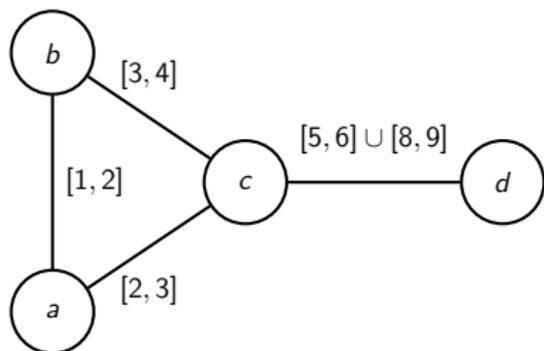
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

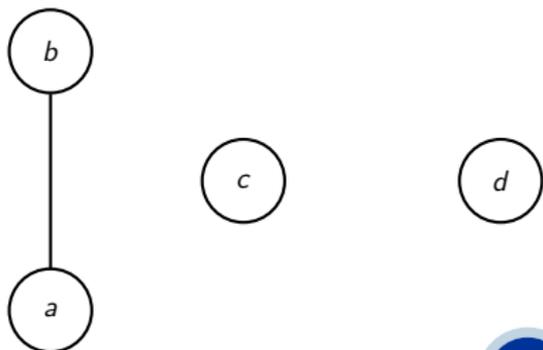
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 1: G_1



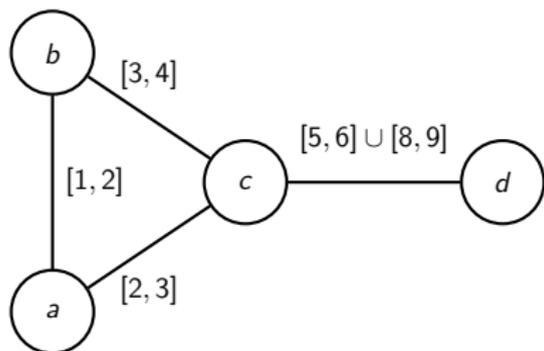
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

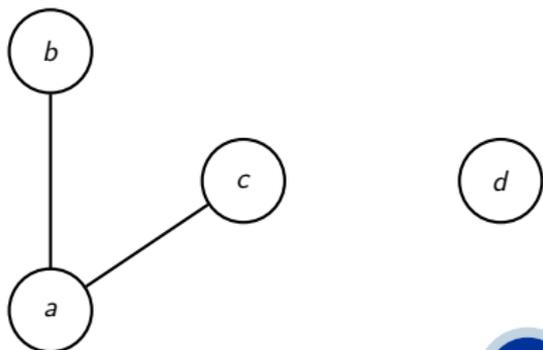
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 2: G_2



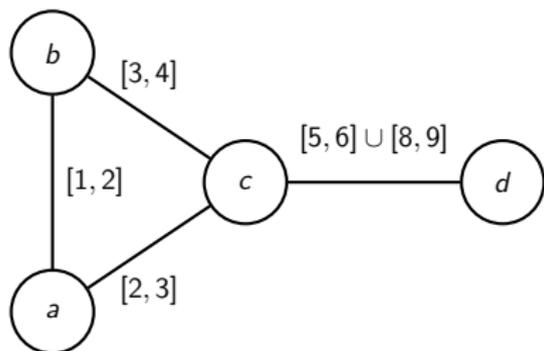
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

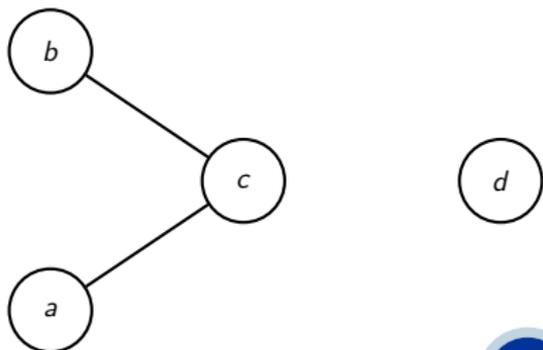
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 3: G_3



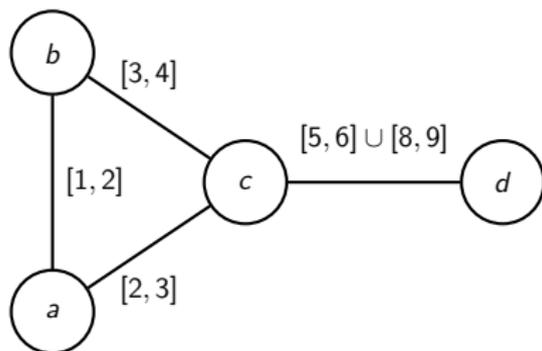
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

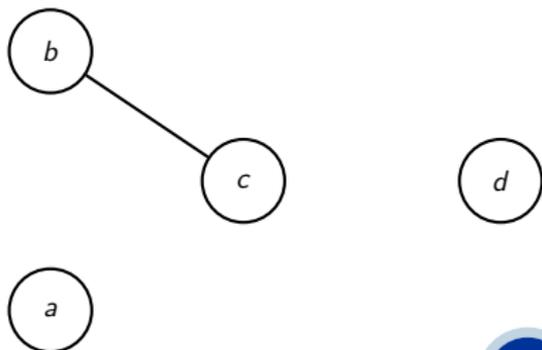
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 4: G_4



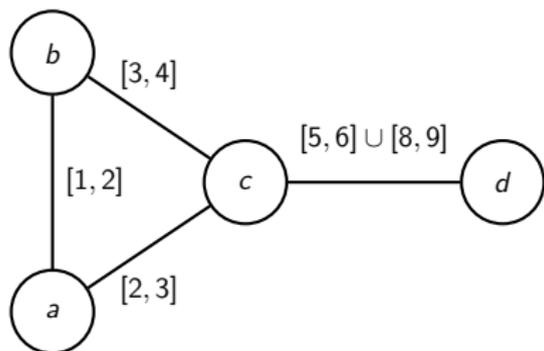
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

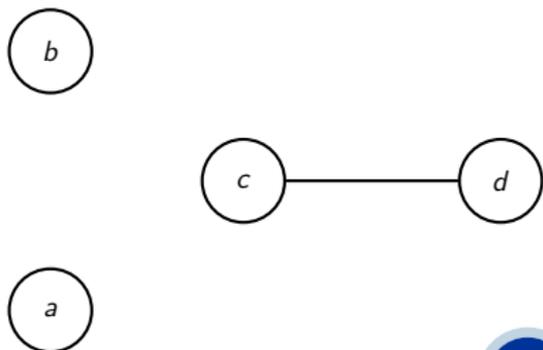
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 5: G_5



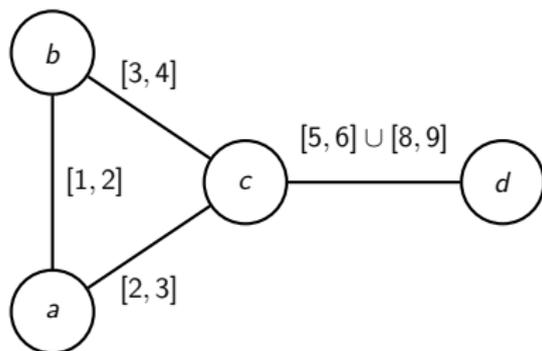
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

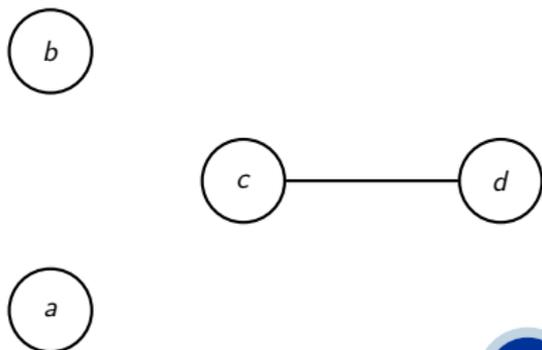
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 6: G_6



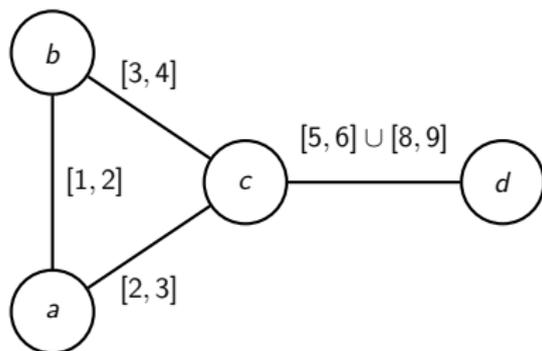
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

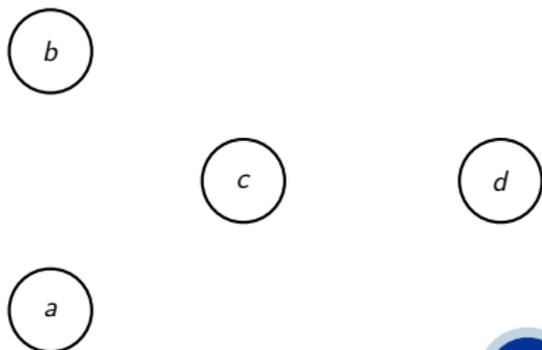
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 7: G_7



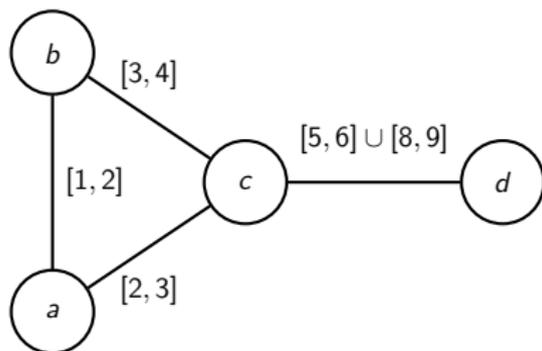
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

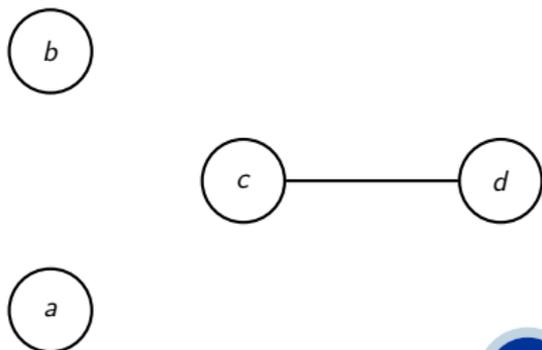
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 8: G_8



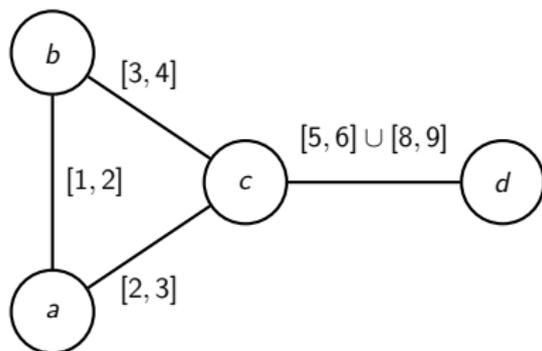
Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

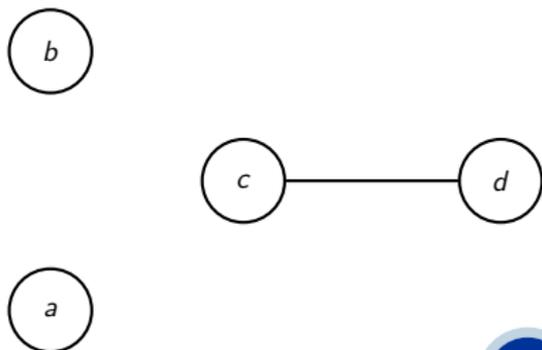
- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$



Snapshot at 9: G_9



Time-Varying Graphs (TVG)

TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$

- V is a set of n nodes
- E is a set of arcs
- \mathcal{T} is an interval over \mathbb{N}^*
- $\rho : E \times \mathcal{T} \rightarrow \{0, 1\}$ is the *presence* function

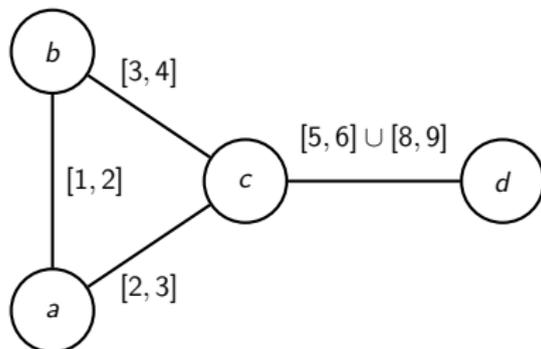
Snapshot of \mathcal{G} at time $t \in \mathcal{T}$:

$$G_t = (V, \{e \in E : \rho(e, t) = 1\})$$

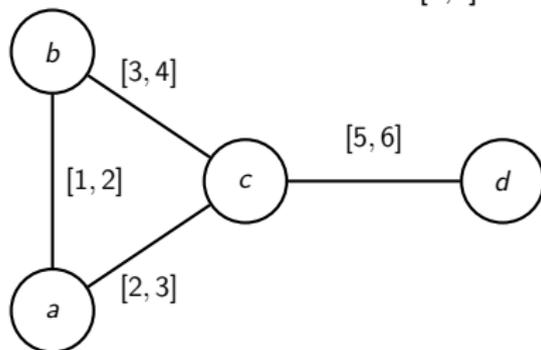
Temporal Subgraph of \mathcal{G} for $[t, t'] \subseteq \mathcal{T}$:

$$\mathcal{G}_{[t, t']} = (V, E, [t, t'], \rho')$$

where ρ' is ρ restricted to $[t, t']$.



Temporal Subgraph $\mathcal{G}_{[1,6]}$:



Journey:

$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$

such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

Temporal length: $t_k - t_1 + 1$

Journey:

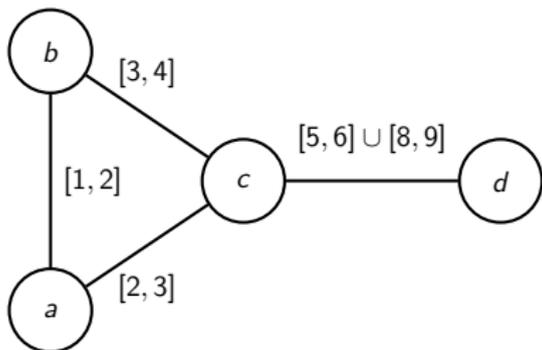
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a
journey of length **5** from a to d



Journey:

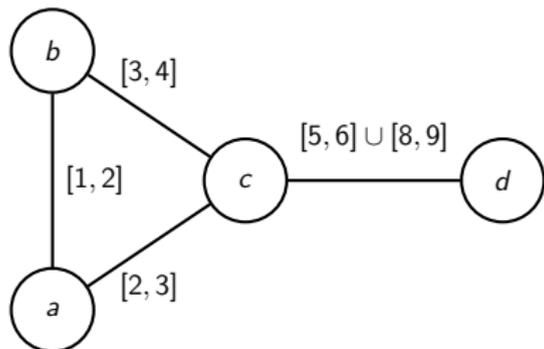
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

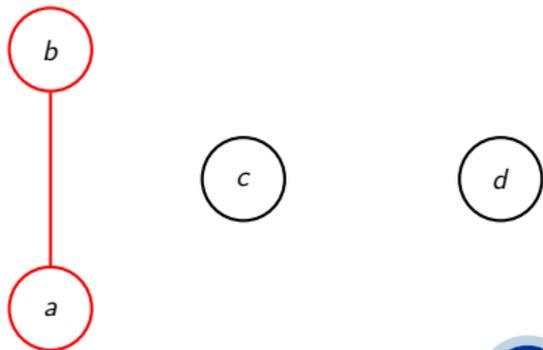
Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length 5 from a to d



Snapshot at time 1: G_1



Journey:

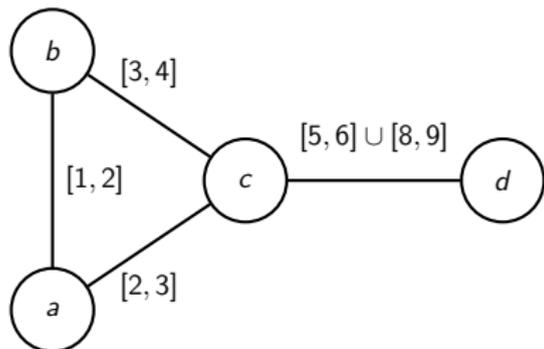
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

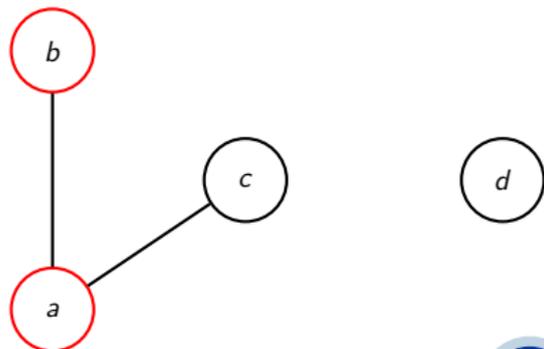
Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 2: G_2



Journey:

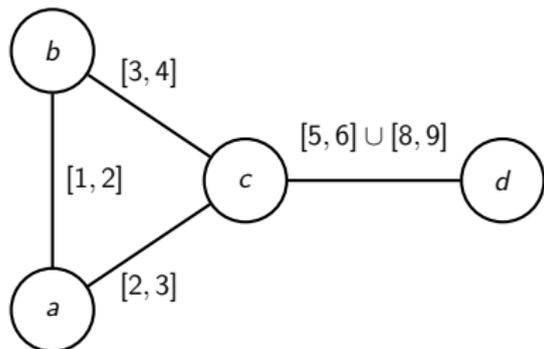
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

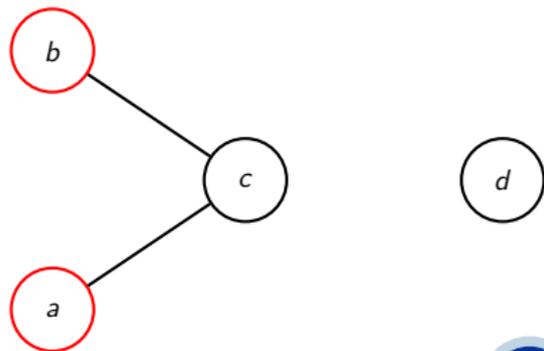
Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 3: G_3



Journey:

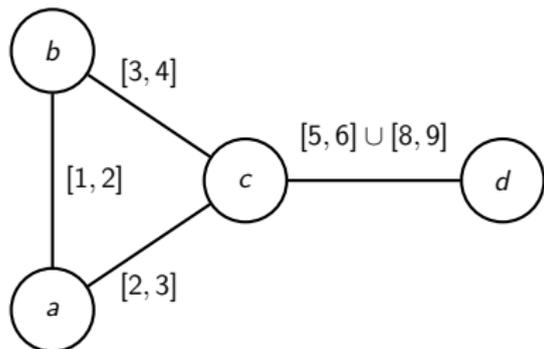
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

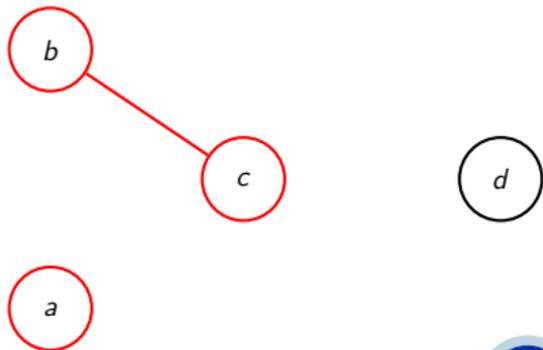
Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 4: G_4



Journey:

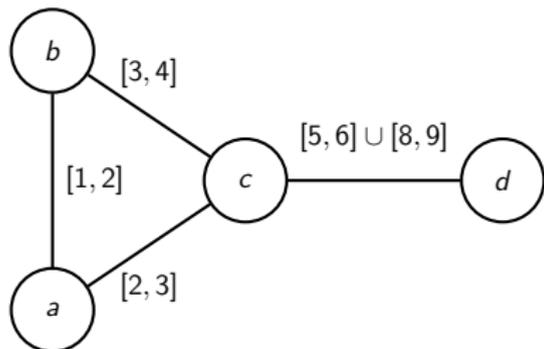
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 5: G_5



Journey:

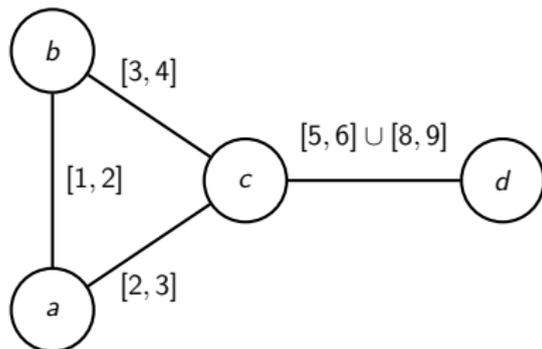
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

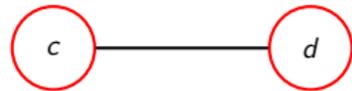
Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 6: G_6



Journey:

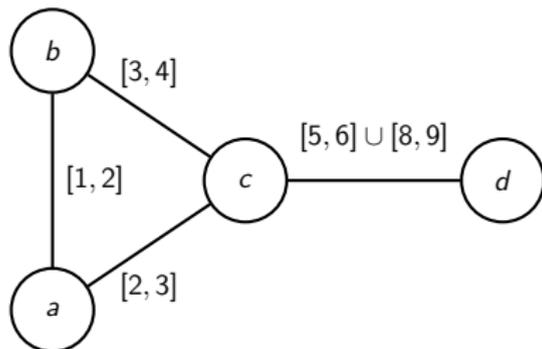
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

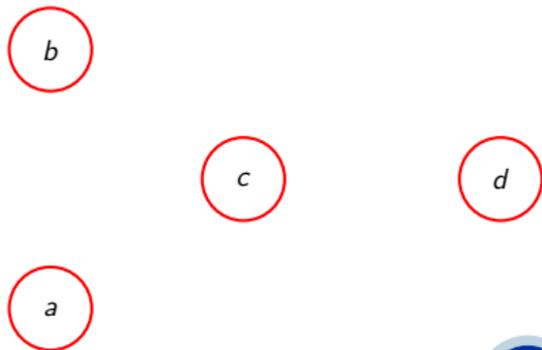
Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 7: G_7



Journey:

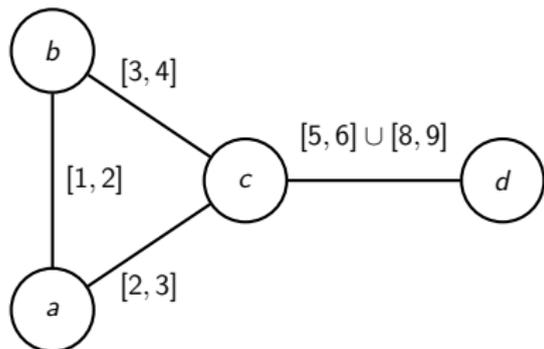
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 8: G_8



Journey:

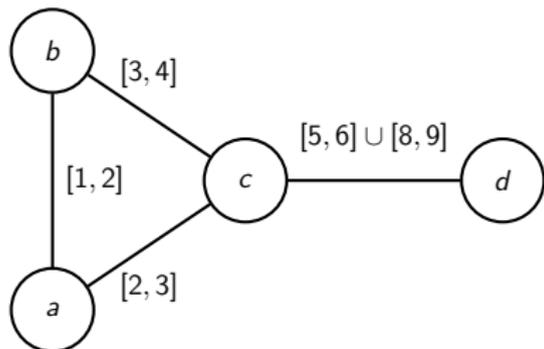
$\mathcal{J} = (e_1, t_1), (e_2, t_2), \dots, (e_k, t_k)$
such that $\forall i \in \{1, \dots, k\}$

- $e_i = (p_i, q_i) \in E$
- $\rho(e_i, t_i) = 1$
- $i < k \Rightarrow q_i = p_{i+1} \wedge t_i < t_{i+1}$

Temporal length: $t_k - t_1 + 1$

Example:

$((a, b), 1), ((b, c), 4), ((c, d), 5)$ is a journey of length **5** from a to d



Snapshot at time 9: G_9



Considered Classes of TVGs

We only consider infinite TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$: \mathcal{T} is **right-open**.

Class $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Bounded Temporal Diameter): At any point in time, every node can reach all the others through a journey of temporal length at most Δ , *i.e.*, **the temporal diameter is bounded by Δ** , [Gómez-Calzado *et al.*, Euro-Par'15]

Class $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Quasi Bounded Temporal Diameter): Every node can always eventually reach each other node through a journey of temporal length at most Δ . [New]

Class $\mathcal{TC}^{\mathcal{R}}$ (Recurrent Temporal Connectivity): At any point in time, every node can reach all the others through a journey, [Casteigts *et al.*, IJPEDES, 2012]

Considered Classes of TVGs

We only consider infinite TVG: $\mathcal{G} = (V, E, \mathcal{T}, \rho)$: \mathcal{T} is **right-open**.

Class $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Bounded Temporal Diameter): At any point in time, every node can reach all the others through a journey of temporal length at most Δ , *i.e.*, **the temporal diameter is bounded by Δ** , [Gómez-Calzado *et al.*, Euro-Par'15]

Class $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Quasi Bounded Temporal Diameter): Every node can always eventually reach each other node through a journey of temporal length at most Δ . [New]

Class $\mathcal{TC}^{\mathcal{R}}$ (Recurrent Temporal Connectivity): At any point in time, every node can reach all the others through a journey, [Casteigts *et al.*, IJPEDES, 2012]

$$\mathcal{TC}^{\mathcal{B}}(\Delta) \subseteq \mathcal{TC}^{\mathcal{Q}}(\Delta) \subseteq \mathcal{TC}^{\mathcal{R}}$$

Computational Model: Synchronous Rounds (1/2)

[Charron-Bost and Moran, STACS'18 / Barjon *et al.*, CJ, 2019]

We model the dynamic network topology by a TVG $\mathcal{G} = (V, E, \mathcal{T}, \rho)$, where \mathcal{T} is **right-open**, *i.e.*, \mathcal{G} is infinite.

Let $\sigma_{\mathcal{T}} = \min \mathcal{T}$ the first instant

The neighborhood of node p at Round i is

$$\mathcal{N}(p)^i = \{q \in V : \rho((p, q), \sigma_{\mathcal{T}} + i - 1) = 1\}$$

(*i.e.*, neighbors of p in the snapshot $G_{\sigma_{\mathcal{T}}+i-1}$)

$\mathcal{N}(p)^i$ is unknown by all nodes

Computational Model: Synchronous Rounds (2/2)

[Charron-Bost and Moran, STACS'18 / Barjon *et al.*, CJ, 2019]

Execution in \mathcal{G} : infinite sequence of configurations $\gamma_0, \gamma_1, \dots$ such that

- γ_0 is arbitrary
- $\forall i > 0$, γ_i is obtained from γ_{i-1} as follows:
 - 1 Every node p sends a message consisting of all or a part of its local state in γ_{i-1} ,
 - 2 p receives all messages sent by nodes in $\mathcal{N}(p)^i$, and
 - 3 p computes its state in γ_i .

$\forall i > 0$,

- γ_{i-1} is the configuration **at the beginning of Round i**
- γ_i is the configuration **at the end of Round i**

Self-stabilization in an Highly Dynamic Context

Adaptation of the definition in the book [Self-Stabilization, Dolev, 2000]

An algorithm \mathcal{A} is **self-stabilizing** for the specification SP on the TVG class \mathcal{C} if

there exists a non-empty subset of configurations \mathcal{L} , called the set of *legitimate configurations*, such that:

- 1 for every $\mathcal{G} \in \mathcal{C}$, for every configuration γ , every execution of \mathcal{A} in \mathcal{G} starting from γ contains a legitimate configuration $\gamma' \in \mathcal{L}$ (**Convergence**), and
- 2 for every $\mathcal{G} \in \mathcal{C}$, for every $t \geq \sigma_{\mathcal{T}}$, for every legitimate configuration $\gamma \in \mathcal{L}$, for every execution e in $\mathcal{G}_{[t, +\infty)}$ starting from γ , $SP(e)$ holds (**Correctness**).

These three classes are **recurring** in the sense that

$$\forall \mathcal{G} \in \mathcal{C}, \forall t \geq o_{\mathcal{T}}, \mathcal{G}_{[t, +\infty)} \in \mathcal{C}$$

In this case, the definition can be simplified as follows

An algorithm \mathcal{A} is **self-stabilizing** for the specification SP on the recurring TVG class \mathcal{C} if

there exists a non-empty subset of legitimate configurations \mathcal{L} such that:

- 1 for every $\mathcal{G} \in \mathcal{C}$, for every configuration γ , every execution of \mathcal{A} in \mathcal{G} starting from γ contains a legitimate configuration $\gamma' \in \mathcal{L}$ (**Convergence**), and
- 2 for every $\mathcal{G} \in \mathcal{C}$, for every legitimate configuration $\gamma \in \mathcal{L}$, for every execution e in \mathcal{G} starting from γ , $SP(e)$ holds (**Correctness**).

Self-stabilizing Algorithms for Highly Dynamic Systems (1/3)

Case-Study: Leader Election

In Classes:

Class $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$: Bounded Temporal Diameter

Class $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ with $\Delta \in \mathbb{N}^*$: Quasi Bounded Temporal Diameter

Class $\mathcal{TC}^{\mathcal{R}}$: Recurrent Temporal Connectivity

$$\mathcal{TC}^{\mathcal{B}}(\Delta) \subseteq \mathcal{TC}^{\mathcal{Q}}(\Delta) \subseteq \mathcal{TC}^{\mathcal{R}}$$

Self-stabilizing Algorithms for Highly Dynamic Systems (2/3)

Case-Study: Leader Election

In Class $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Bounded Temporal Diameter)

- Δ known
- Stabilization Time: at most 3Δ rounds
- Memory Requirement: $O(\log(n + \Delta))$ bits per node

Self-stabilizing Algorithms for Highly Dynamic Systems (3/3)

Case-Study: Leader Election

Class $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Quasi Bounded Temporal Diameter)

- Δ and n known
- Memory Requirement: $O(n(\log(n + \Delta)))$ bits per node

Class $\mathcal{TC}^{\mathcal{R}}$ (Recurrent Temporal Connectivity)

- n known
- Memory Requirement: infinite

Self-stabilizing Algorithms for Highly Dynamic Systems (3/3)

Case-Study: Leader Election

Class $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Quasi Bounded Temporal Diameter)

- Δ and n known
- Memory Requirement: $O(n(\log(n + \Delta)))$ bits per node

Class $\mathcal{TC}^{\mathcal{R}}$ (Recurrent Temporal Connectivity)

- n known
- Memory Requirement: infinite

Stabilization time unboundable in those two classes, but ...

Speculation [Kotla *et al.*, ACM Trans. Comput. Syst., 2009]:

- the system satisfies its requirements for all executions,
- but also exhibits significantly better performances in a subset of more probable executions.

Speculation [Kotla *et al.*, ACM Trans. Comput. Syst., 2009]:

- the system satisfies its requirements for all executions,
- but also exhibits significantly better performances in a subset of more probable executions.

Idea: worst possible scenarios are often rare in practice.

A speculative algorithm self-adapts its performances *w.r.t.* the “quality” of the environment, *i.e.*, the more favorable the environment is, the better the complexity of the algorithm should be.

Speculation [Kotla *et al.*, ACM Trans. Comput. Syst., 2009]:

- the system satisfies its requirements for all executions,
- but also exhibits significantly better performances in a subset of more probable executions.

Idea: worst possible scenarios are often rare in practice.

A speculative algorithm self-adapts its performances *w.r.t.* the “quality” of the environment, *i.e.*, the more favorable the environment is, the better the complexity of the algorithm should be.

In **Self-Stabilizing (Static) Systems**: a self-stabilizing mutual exclusion algorithm whose stabilization time is significantly better when the execution is synchronous. [Dubois and Guerraoui, PODC'13]

Speculative Self-stabilizing Algorithms for Highly Dynamic Systems

Our solutions

Class $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (Quasi Bounded Temporal Diameter)

- Δ and n known
- Memory Requirement: $O(n(\log(n + \Delta)))$ bits per node
- **Speculation:** stabilization time in $\mathcal{TC}^{\mathcal{B}}(\Delta) \subseteq \mathcal{TC}^{\mathcal{Q}}(\Delta)$ is at most 2Δ rounds

Class $\mathcal{TC}^{\mathcal{R}}$ (Recurrent Temporal Connectivity)

- n known
- Memory Requirement: infinite
- **Speculation:** stabilization time in $\mathcal{TC}^{\mathcal{B}}(\Delta) \subseteq \mathcal{TC}^{\mathcal{R}}$ is at most $\Delta + 1$ rounds

Overview of our solutions

Nodes are **identified**: $\forall p \in V$, $id(p)$ is unique identifier of p

Overview of our solutions

Nodes are **identified**: $\forall p \in V$, $id(p)$ is unique identifier of p

Let $IDSET$ be the definition domain of the identifiers
(*n.b.*, usually $|IDSET| \gg n$)

Overview of our solutions

Nodes are **identified**: $\forall p \in V$, $id(p)$ is unique identifier of p

Let $IDSET$ be the definition domain of the identifiers
(*n.b.*, usually $|IDSET| \gg n$)

$\forall v \in IDSET$,

- v is a **real ID** if $\exists p \in V$, $id(p) = v$,
- v is a **fake ID** otherwise

Overview of our solutions

Nodes are **identified**: $\forall p \in V$, $id(p)$ is unique identifier of p

Let $IDSET$ be the definition domain of the identifiers
(*n.b.*, usually $|IDSET| \gg n$)

$\forall v \in IDSET$,

- v is a **real ID** if $\exists p \in V, id(p) = v$,
- v is a **fake ID** otherwise

Every node p computes the identifier of the leader in $lid(p)$
Initially, the value of $lid(p)$ may be a fake ID

Overview of our solutions

Nodes are **identified**: $\forall p \in V$, $id(p)$ is unique identifier of p

Let $IDSET$ be the definition domain of the identifiers
(*n.b.*, usually $|IDSET| \gg n$)

$\forall v \in IDSET$,

- v is a **real ID** if $\exists p \in V$, $id(p) = v$,
- v is a **fake ID** otherwise

Every node p computes the identifier of the leader in $lid(p)$
Initially, the value of $lid(p)$ may be a fake ID

Strategy:

- 1 First, **eliminate all fake IDs**, and then
- 2 Compute in all output variables the **minimum real ID**, noted $id(\ell)$.

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (1/3)

Bounded Temporal Diameter, Δ known

Variables: $lid(p) \in IDSET$ and $t(p) \in \{0, \dots, 2\Delta\}$

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (1/3)

Bounded Temporal Diameter, Δ known

Variables: $lid(p) \in IDSET$ and $t(p) \in \{0, \dots, 2\Delta\}$

For each node p , at each round:

- 1 p sends $\langle lid(p), t(p) \rangle$
- 2 If p receives some messages, then
 - ▶ Let (x, t_x) the smallest received pair (lexicographic order)
 - ▶ if $\langle x, t_x \rangle < \langle lid(p), t(p) \rangle$, then $\langle lid(p), t(p) \rangle := \langle x, t_x \rangle$
- 3 $t(p) ++$
- 4 if $lid(p) \geq id(p) \vee t \geq 2\Delta$, then $\langle lid(p), t(p) \rangle := \langle id(p), 0 \rangle$ (Reset)

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (2/3)

Bounded Temporal Diameter, Δ known

Legitimate Configurations:

$$\forall p \in V, lid(p) = id(\ell) \wedge t(p) \leq \Delta \wedge p = \ell \Rightarrow t(p) = 0$$

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (2/3)

Bounded Temporal Diameter, Δ known

Legitimate Configurations:

$$\forall p \in V, lid(p) = id(\ell) \wedge t(p) \leq \Delta \wedge p = \ell \Rightarrow t(p) = 0$$

Correctness:

- No fake ID $\Rightarrow id(\ell)$ is the minimum value of *IDSET* in the network.

So, $lid(\ell) = id(\ell)$ and $t(\ell) = 0$ forever

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (2/3)

Bounded Temporal Diameter, Δ known

Legitimate Configurations:

$$\forall p \in V, lid(p) = id(\ell) \wedge t(p) \leq \Delta \wedge p = \ell \Rightarrow t(p) = 0$$

Correctness:

- No fake ID $\Rightarrow id(\ell)$ is the minimum value of *IDSET* in the network.

So, $lid(\ell) = id(\ell)$ and $t(\ell) = 0$ forever

- Bounded temporal diameter \Rightarrow no reset

So, $\forall p \in V, lid(p) = id(\ell)$ forever

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (3/3)

Bounded Temporal Diameter, Δ known

Convergence:

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (3/3)

Bounded Temporal Diameter, Δ known

Convergence:

- The timestamps associated to each fake ID increment at each round until reaching 2Δ and so vanishing,

i.e., after at most 2Δ rounds, no fake ID

In particular, $lid(\ell) = id(\ell)$ and $t(\ell) = 0$ forever

Self-stabilization in $\mathcal{TC}^{\mathcal{B}}(\Delta)$ with $\Delta \in \mathbb{N}^*$ (3/3)

Bounded Temporal Diameter, Δ known

Convergence:

- The timestamps associated to each fake ID increment at each round until reaching 2Δ and so vanishing,

i.e., after at most 2Δ rounds, no fake ID

In particular, $lid(\ell) = id(\ell)$ and $t(\ell) = 0$ forever

- At most Δ additional rounds are necessary to reach a configuration where $\forall p \in V, lid(p) = id(p) \vee t(p) \leq \Delta$

Hence, a stabilization time of at most 3Δ rounds.

Self-stabilization in $\mathcal{TC}^{\mathcal{Q}}(\Delta)$ with $\Delta \in \mathbb{N}^*$

Quasi Bounded Temporal Diameter, n and Δ known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: queue of **at most n** pairs $\langle id, t \rangle \in IDSET \times \{0, \dots, \Delta\}$

Self-stabilization in $\mathcal{TC}^Q(\Delta)$ with $\Delta \in \mathbb{N}^*$

Quasi Bounded Temporal Diameter, n and Δ known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: queue of **at most** n pairs $\langle id, t \rangle \in IDSET \times \{0, \dots, \Delta\}$

For each node p , at each round:

- p sends every pair $\langle x, y \rangle \in members(p)$ such that $y < \Delta$

Self-stabilization in $\mathcal{TC}^Q(\Delta)$ with $\Delta \in \mathbb{N}^*$

Quasi Bounded Temporal Diameter, n and Δ known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: queue of **at most** n pairs $\langle id, t \rangle \in IDSET \times \{0, \dots, \Delta\}$

For each node p , at each round:

- p sends every pair $\langle x, y \rangle \in members(p)$ such that $y < \Delta$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then
replace the timestamp by y if y is smaller
 - ▶ Otherwise, if $members(p)$ is full, remove the tail
Insert $\langle x, y \rangle$ at the head

Self-stabilization in $\mathcal{TC}^Q(\Delta)$ with $\Delta \in \mathbb{N}^*$

Quasi Bounded Temporal Diameter, n and Δ known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: queue of **at most** n pairs $\langle id, t \rangle \in IDSET \times \{0, \dots, \Delta\}$

For each node p , at each round:

- p sends every pair $\langle x, y \rangle \in members(p)$ such that $y < \Delta$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then
replace the timestamp by y if y is smaller
 - ▶ Otherwise, if $members(p)$ is full, remove the tail
Insert $\langle x, y \rangle$ at the head
- Increment all timestamps in $members(p)$

Self-stabilization in $\mathcal{TC}^Q(\Delta)$ with $\Delta \in \mathbb{N}^*$

Quasi Bounded Temporal Diameter, n and Δ known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: queue of **at most** n pairs $\langle id, t \rangle \in IDSET \times \{0, \dots, \Delta\}$

For each node p , at each round:

- p sends every pair $\langle x, y \rangle \in members(p)$ such that $y < \Delta$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then
replace the timestamp by y if y is smaller
 - ▶ Otherwise, if $members(p)$ is full, remove the tail
Insert $\langle x, y \rangle$ at the head
- Increment all timestamps in $members(p)$
- if $members(p)$ is full, remove the tail
Insert $\langle id(p), 0 \rangle$

Self-stabilization in $\mathcal{TC}^Q(\Delta)$ with $\Delta \in \mathbb{N}^*$

Quasi Bounded Temporal Diameter, n and Δ known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: queue of **at most** n pairs $\langle id, t \rangle \in IDSET \times \{0, \dots, \Delta\}$

For each node p , at each round:

- p sends every pair $\langle x, y \rangle \in members(p)$ such that $y < \Delta$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then
replace the timestamp by y if y is smaller
 - ▶ Otherwise, if $members(p)$ is full, remove the tail
Insert $\langle x, y \rangle$ at the head
- Increment all timestamps in $members(p)$
- if $members(p)$ is full, remove the tail
Insert $\langle id(p), 0 \rangle$
- Update $lid(p)$ with the smallest ID in $members(p)$

Self-stabilization in $\mathcal{TC}^Q(\Delta)$ with $\Delta \in \mathbb{N}^*$

Quasi Bounded Temporal Diameter, n and Δ known (2/3)

Legitimate Configurations: $\forall p \in V,$
 $lid(p) = id(\ell) \wedge \{id : \langle id, _ \rangle \in members(p)\} = \{id(q) : q \in V\}$

Correctness: trivial since the set of legitimate configuration is closed

Convergence:

- The timestamps associated to each fake ID increment at each round until reaching Δ : **after at most Δ rounds**, no fake ID is sent and so **no fake ID can be ever inserted in a *members* queue**.

Convergence:

- The timestamps associated to each fake ID increment at each round until reaching Δ : **after at most Δ rounds**, no fake ID is sent and so **no fake ID can be ever inserted in a *members* queue.**
- Quasi Bounded Temporal Diameter \Rightarrow **every real ID is regularly inserted in each *members* queue.**

Convergence:

- The timestamps associated to each fake ID increment at each round until reaching Δ : **after at most Δ rounds**, no fake ID is sent and so **no fake ID can be ever inserted in a *members* queue**.
- Quasi Bounded Temporal Diameter \Rightarrow **every real ID is regularly inserted in each *members* queue**.

Speculation:

Bounded Temporal Diameter \Rightarrow **all real ID are inserted in each *members* queue in each period of Δ rounds**

Hence, the stabilization time is **at most 2Δ rounds** in $\mathcal{TC}^B(\Delta)$.

Self-stabilization in $\mathcal{TC}^{\mathcal{R}}$

Recurrent Temporal Connectivity, n known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: map of **at most** n pairs $\langle id, t \rangle \in IDSET \times \mathbb{N}$

Self-stabilization in $\mathcal{TC}^{\mathcal{R}}$

Recurrent Temporal Connectivity, n known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: map of **at most n** pairs $\langle id, t \rangle \in IDSET \times \mathbb{N}$

For each node p , at each round:

- p sends $members(p)$

Self-stabilization in \mathcal{TC}^R

Recurrent Temporal Connectivity, n known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: map of **at most** n pairs $\langle id, t \rangle \in IDSET \times \mathbb{N}$

For each node p , at each round:

- p sends $members(p)$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then replace the timestamp by y if y is smaller
 - ▶ Otherwise, let t be the largest timestamp in $members(p)$
If $y < t$ and $members(p)$ is full, then remove a pair $\langle _, t \rangle$ from $members$
If $members(p)$ is not full, then Insert $\langle x, y \rangle$ in $members(p)$

Self-stabilization in \mathcal{TC}^R

Recurrent Temporal Connectivity, n known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: map of **at most** n pairs $\langle id, t \rangle \in IDSET \times \mathbb{N}$

For each node p , at each round:

- p sends $members(p)$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then replace the timestamp by y if y is smaller
 - ▶ Otherwise, let t be the largest timestamp in $members(p)$
If $y < t$ and $members(p)$ is full, then remove a pair $\langle _, t \rangle$ from $members$
If $members(p)$ is not full, then Insert $\langle x, y \rangle$ in $members(p)$
- Increment all timestamps in $members(p)$

Self-stabilization in \mathcal{TC}^R

Recurrent Temporal Connectivity, n known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: map of **at most n** pairs $\langle id, t \rangle \in IDSET \times \mathbb{N}$

For each node p , at each round:

- p sends $members(p)$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then replace the timestamp by y if y is smaller
 - ▶ Otherwise, let t be the largest timestamp in $members(p)$
If $y < t$ and $members(p)$ is full, then remove a pair $\langle _, t \rangle$ from $members$
If $members(p)$ is not full, then Insert $\langle x, y \rangle$ in $members(p)$
- Increment all timestamps in $members(p)$
- If $members(p)$ is full, then remove a pair in $members$ with a largest timestamp
Insert $\langle id(p), 0 \rangle$

Self-stabilization in \mathcal{TC}^R

Recurrent Temporal Connectivity, n known (1/3)

Variables:

- $lid(p) \in IDSET$
- $members(p)$: map of **at most n** pairs $\langle id, t \rangle \in IDSET \times \mathbb{N}$

For each node p , at each round:

- p sends $members(p)$
- For each received pairs $\langle x, y \rangle$
 - ▶ If x already appears in a pair of $members(p)$, then replace the timestamp by y if y is smaller
 - ▶ Otherwise, let t be the largest timestamp in $members(p)$
If $y < t$ and $members(p)$ is full, then remove a pair $\langle _, t \rangle$ from $members$
If $members(p)$ is not full, then Insert $\langle x, y \rangle$ in $members(p)$
- Increment all timestamps in $members(p)$
- If $members(p)$ is full, then remove a pair in $members$ with a largest timestamp
Insert $\langle id(p), 0 \rangle$
- Update $lid(p)$ with the smallest ID in $members(p)$

Legitimate Configurations: $\forall p \in V,$

$$lid(p) = id(\ell) \wedge \{id : \langle id, _ \rangle \in members(p)\} = \{id(q) : q \in V\}$$

Correctness: trivial since the set of legitimate configuration is closed

Convergence: similar to the previous algorithm

Speculation: in $\mathcal{TC}^B(\Delta)$,

- After the first round, at each round, each node p sends $\langle id(p), 0 \rangle$
- For every node q , a pair $\langle id(p), x_q \rangle$ with $x_q \leq \Delta$ reaches q within x rounds.
- At that time, all fake IDs have a timestamps $> x_q$.
So, $id(p)$ is inserted and never more removed.

Hence, a stabilization time **in at most $\Delta + 1$ rounds** in $\mathcal{TC}^B(\Delta)$.

It is a first attempt ...

It is a first attempt ...

Yet, we have circumvented the impossibility result of [Braud-Santoni *et al.*, IJNC, 2016] by considering non-silent solutions

Actually, we have even considered **more general classes!**

It is a first attempt ...

Yet, we have circumvented the impossibility result of [Braud-Santoni *et al.*, IJNC, 2016] by considering non-silent solutions

Actually, we have even considered **more general classes!**

Extensions:

- Even more general classes
- Expressiveness in particular TVG classes

Transformer (e.g., [Katz and Perry, DC, 1993]):
Propagation of Information with Feedback + Leader Election

- Karine Altisen, Stéphane Devismes, and Anaïs Durand, Colette Johnen, and Franck Petit. *Self-stabilizing Systems in Spite of High Dynamics*. *Submitted to OPODIS'19*.