# Characterizing Topological Assumptions of Distributed Algorithms in Dynamic Networks

## A.Casteigts, S.Chaumette, A.Ferreira

June 1, 2009

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

(日) (同) (三) (三)

Introduction	Purpose	Applications	Conclusion
000			
Distributed Algor	ithms		

- Local interactions only.
- Abstraction of the communication model.

A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusio
•••	00	000000	
Distributed Algor	rithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).

3

・ロッ ・回 ・ ・ ヨ ・ ・

Introduction	Purpose	Applications	Co
●○○	00	000000	
Distributed Ala	orithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).

#### Example (spanning tree with pre-selected root) Ν N 0 0 initial states: I (the root), N (all others), 0 (edges) 0 0 0 relabelling rule: 0 0 Ν Ň Ν < (D) >

#### A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Co
●○○	00	000000	
Distributed Ala	orithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).

#### Example (spanning tree with pre-selected root) Ν N 0 0 initial states: I (the root), N (all others), 0 (edges) 0 0 0 relabelling rule: 0 0 Ν Ň Ν < (D) >

#### A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Co
●○○	00	000000	
Distributed Ala	orithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).



#### A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Co
●○○	00	000000	
Distributed Ala	orithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).

## Example (spanning tree with pre-selected root) initial states: I (the root), N (all others), 0 (edges) relabelling rule: $\stackrel{I}{\longrightarrow} \stackrel{O}{\longrightarrow} \stackrel{N}{\longrightarrow} \stackrel{I}{\longrightarrow} \stackrel{I}{\rightarrow} \stackrel{I}{\longrightarrow} \stackrel{I}{\rightarrow} \stackrel$

#### A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Co
●○○	00	000000	
Distributed Alg	orithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).

## Example (spanning tree with pre-selected root) initial states: I (the root), N (all others), 0 (edges) relabelling rule: $\stackrel{I}{\longrightarrow} \stackrel{O}{\longrightarrow} \stackrel{N}{\longrightarrow} \stackrel{I}{\longrightarrow} \stackrel{I}{\rightarrow} \stackrel{I}{\longrightarrow} \stackrel{I}{\rightarrow} \stackrel$

#### A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Co
●○○	00	000000	
Distributed Ala	orithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).

## Example (spanning tree with pre-selected root) initial states: I (the root), N (all others), 0 (edges) relabelling rule: $\stackrel{I}{\longrightarrow} \stackrel{O}{\longrightarrow} \stackrel{N}{\longrightarrow} \stackrel{I}{\longrightarrow} \stackrel$

#### A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Co
●○○	00	000000	
Distributed Alg	orithms		

- Local interactions only.
- Abstraction of the communication model.

## Graph Relabellings [Litovsky, Métivier, Sopena 1999]

- State of vertices and edges represented by labels.
- Distributed operations are transition patterns (*relabelling rules*) on these labels (*preconditions, actions*).

## Example (spanning tree with pre-selected root)

initial states: I (the root), N (all others), 0 (edges)

• relabelling rule:  $\int_{0}^{1} 0$ 



Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

1

1

0

< (<sup>†</sup>1<sup>1</sup>) ►

0

	Purpose	Applications	Conclusion
Characterization			
Two approach	es		

Characterizing what can be done in a given context

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

= 990

・ロン ・四 と ・ ヨ と ・ ヨ と …

Introd	luction	Purpose 00	Applications 0000000	Conclusion
Cha	aracterization			
	Two approaches			
	<ul> <li>Characterizing w</li> </ul>	/hat can be do	one in a given context	

Characterizing in what context a given thing can be done

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

2

・ロン ・四 と ・ ヨ と ・ ヨ と …

Introduction	Purpose	Applications	Conclusion
○●○	00	0000000	
Characterization			

## Two approaches

- Characterizing what can be done in a given context
  - e.g. what can be done in a complete (or arborescent) interaction graphs



Characterizing in what context a given thing can be done

э

・ロト ・回ト ・ 回ト ・

Introduction	Purpose	Applications	Conclusion
○●○	00	0000000	
Characterization			

## Two approaches

- Characterizing what can be done in a given context
  - e.g. what can be done in a complete (or arborescent) interaction graphs



Characterizing in what context a given thing can be done

э

・ロト ・回ト ・ 回ト ・

Introduction	Purpose	Applications	Conclusion
○●○	00	0000000	
Characterization			

## Two approaches

- Characterizing what can be done in a given context
  - *e.g.* what can be done in a complete (or arborescent) interaction graphs



- Characterizing in what context a given thing can be done
  - what properties a given algorithm requires on the graph dynamics? (*i.e.*, on the *topological* evolution)

э

(日) (同) (三) (

Introduction	Purpose	Applications	Conclusion
000			
Intuitive example			

- initial states: I for the initial emitter, N for all the other nodes
- relabelling rule :  $\stackrel{I}{\bullet} \stackrel{N}{\longrightarrow} \stackrel{I}{\longrightarrow} \stackrel{I}{\bullet} \stackrel{I}{\longrightarrow}$

A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000	00	0000000	
Intuitive example			

initial states: I for the initial emitter, N for all the other nodes



A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000	00	0000000	
Intuitive example			

■ initial states: I for the initial emitter, N for all the other nodes



A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000			
Intuitive example			

initial states: I for the initial emitter, N for all the other nodes



▲ロト ▲圖 ▶ ▲ 臣 ▶ ▲ 臣 ▶ ○ 臣 = の � @

A.Casteigts, S.Chaumette, A.Ferreira



initial states: I for the initial emitter, N for all the other nodes



A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000			
Intuitive example			

initial states: I for the initial emitter, N for all the other nodes



A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000			
Intuitive example			

initial states: I for the initial emitter, N for all the other nodes



A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000	●○	0000000	
Formalism to represen	t dynamic topology		



Introduction	Purpose	Applications	Conclusion
000	●○	0000000	
Formalism to represen	t dynamic topology		



Introduction	Purpose	Applications	Conclusion
000	●○	0000000	
Formalism to represen	t dynamic topology		



Introduction OCO Formalism to represent dynamic topology



Introduction OCO Formalism to represent dynamic topology



$$\begin{aligned} \mathcal{S}_{\mathbb{T}} &= t_0, t_1, t_2, t_3, t_4 \\ \mathcal{S}_G &= G_0, G_1, G_2, G_3 \\ \mathcal{G} &= \bigcup_{G_i \in \mathcal{S}_G} = \checkmark \end{aligned}$$

900

Introduction October Purpose Applications Conclusion October Porpose Applications Conclusion October Porpose Purpose October P





 $\downarrow$  graphical representation  $\downarrow$ 



590

#### A.Casteigts, S.Chaumette, A.Ferreira

	Purpose	Applications	Conclusion
	00 <sup>°</sup>		
Formalism to re	present dynamic to	pology	

$$\begin{aligned} \mathcal{S}_{\mathbb{T}} &= t_0, t_1, t_2, t_3, t_4 \\ \mathcal{S}_G &= \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \\ \mathcal{G} &= \bigcup_{\mathcal{G}_i \in \mathcal{S}_G} = \checkmark \end{aligned}$$

$$\mathcal{G} = (\mathcal{G}, \mathcal{S}_{\mathcal{G}}, \mathcal{S}_{\mathbb{T}})$$

is the corresponding Evolving Graph.

 $\downarrow$  graphical representation  $\downarrow$ 



	Purpose	Applications	Conclusion
	•o		
Formalism to repr	resent dynamic to	nology	

$$\begin{aligned} \mathcal{S}_{\mathbb{T}} &= t_0, t_1, t_2, t_3, t_4 \\ \mathcal{S}_{\mathcal{G}} &= \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \\ \mathcal{G} &= \bigcup_{\mathcal{G}_i \in \mathcal{S}_{\mathcal{G}}} = \checkmark \end{aligned}$$

$$\mathcal{G} = (\mathcal{G}, \mathcal{S}_{\mathcal{G}}, \mathcal{S}_{\mathbb{T}})$$

is the corresponding Evolving Graph.

 $\downarrow$  graphical representation  $\downarrow$ 



 $\rightarrow\,$  possibility to express topological properties, and to define related concepts.

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

2

	Purpose	Applications	Conclusion
	00 <sup>°</sup>		
Formalism to repre	esent dynamic to	pology	

$$\begin{aligned} \mathcal{S}_{\mathbb{T}} &= t_0, t_1, t_2, t_3, t_4 \\ \mathcal{S}_G &= \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \\ \mathcal{G} &= \bigcup_{\mathcal{G}_i \in \mathcal{S}_G} = \checkmark \end{aligned}$$

$$\mathcal{G} = (\mathcal{G}, \mathcal{S}_{\mathcal{G}}, \mathcal{S}_{\mathbb{T}})$$

is the corresponding Evolving Graph.

 $\downarrow$  graphical representation  $\downarrow$ 



- $\rightarrow\,$  possibility to express topological properties, and to define related concepts.
  - e.g. Journey (path over time).

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

2

	Purpose	Applications	Conclusion
	•o		
Formalism to re	present dynamic to	nology	

$$\begin{aligned} \mathcal{S}_{\mathbb{T}} &= t_0, t_1, t_2, t_3, t_4 \\ \mathcal{S}_G &= \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \\ \mathcal{G} &= \bigcup_{\mathcal{G}_i \in \mathcal{S}_G} = \checkmark \end{aligned}$$

$$\downarrow$$
 graphical representation  $\downarrow$ 



$$\mathcal{G} = (G, \mathcal{S}_G, \mathcal{S}_{\mathbb{T}})$$

is the corresponding Evolving Graph.

 $\mathcal{J}_{a,e} = \{(a, b, 1), (b, c, 1), (c, d, 1), (d, e, 2)\}$  is a journey from *a* to *e*.

- $\rightarrow\,$  possibility to express topological properties, and to define related concepts.
  - e.g. Journey (path over time).

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

	Purpose	Applications	Conclusior
	00		
Formalism to rep	resent dynamic to	nology	

 $\mathcal{G} = (G, \mathcal{S}_G, \mathcal{S}_{\mathbb{T}})$ 

### Evolving graphs [Ferreira 2004]

$$\begin{aligned} \mathcal{S}_{\mathbb{T}} &= t_0, t_1, t_2, t_3, t_4 \\ \mathcal{S}_G &= \mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 \\ \mathcal{G} &= \bigcup_{\mathcal{G}_i \in \mathcal{S}_G} = \checkmark \end{aligned}$$

 $\downarrow$  graphical representation  $\downarrow$ 



 $\begin{aligned} \mathcal{J}_{a,e} &= \{(a,b,1), (b,c,1), (c,d,1), (d,e,2)\} \\ \text{is a journey from $a$ to $e$.} \\ \mathcal{J}_{a,e} &= \{(a,c,0), (c,e,2)\} \\ \text{is a strict} \\ \text{journey from $a$ to $e$.} \end{aligned}$ 

イロト イポト イヨト イヨト

is the corresponding Evolving Graph.

- $\rightarrow\,$  possibility to express topological properties, and to define related concepts.
  - e.g. Journey (path over time).

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

Introd	

Purpose ○● Applications

Conclusion

## Combination

## Relabellings over Evolving Graphs



◆□ > ◆□ > ◆三 > ◆三 > 三 のへの

A.Casteigts, S.Chaumette, A.Ferreira

Introd	
000	

Purpose ○● Applications

Conclusion

## Combination

## Relabellings over Evolving Graphs



▲□▶ ▲□▶ ▲目▶ ▲目▶ 三回 ろん⊙

A.Casteigts, S.Chaumette, A.Ferreira
#### Combination



An execution is an alternated sequence of relabellings and topological events:  $X = \mathcal{R}_{\mathcal{A}_{[t_{last-1}, t_{last}]}} \circ Event_{t_{last-1}} \circ .. \circ Event_{t_i} \circ \mathcal{R}_{\mathcal{A}_{[t_{i-1}, t_i]}} \circ .. \circ Event_{t_1} \circ \mathcal{R}_{\mathcal{A}_{[t_0, t_1]}}(\mathbf{G}_0)$ 

#### ◆□ > ◆□ > ◆臣 > ◆臣 > □ □ ○ ○ ○

A.Casteigts, S.Chaumette, A.Ferreira

#### Combination



An execution is an alternated sequence of relabellings and topological events:  $X = \mathcal{R}_{\mathcal{A}_{[t_{last-1}, t_{last}]}} \circ Event_{t_{last-1}} \circ .. \circ Event_{t_i} \circ \mathcal{R}_{\mathcal{A}_{[t_{i-1}, t_i]}} \circ .. \circ Event_{t_1} \circ \mathcal{R}_{\mathcal{A}_{[t_0, t_1]}}(\mathbf{G}_0)$ We note  $\mathcal{X}_{\mathcal{A}/\mathcal{G}}$  the set of all possible execution sequences of an algorithm  $\mathcal{A}$  over an evolving graph  $\mathcal{G}$ 

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

<ロ> <同> <同> < 回> < 回>

#### Combination



An execution is an alternated sequence of relabellings and topological events:  $X = \mathcal{R}_{\mathcal{A}_{[t_{l_{ast}-1}, t_{l_{ast}}]}} \circ Event_{t_{l_{ast}-1}} \circ .. \circ Event_{t_i} \circ \mathcal{R}_{\mathcal{A}_{[t_{i-1}, t_i]}} \circ .. \circ Event_{t_1} \circ \mathcal{R}_{\mathcal{A}_{[t_0, t_1]}}(\mathbf{G}_0)$ We note  $\mathcal{X}_{\mathcal{A}/\mathcal{G}}$  the set of all possible execution sequences of an algorithm  $\mathcal{A}$ over an evolving graph  $\mathcal{G}$ 

Topology-related necessary condition:  $\neg C_{\mathcal{N}}(\mathcal{G}) \implies \nexists X \in \mathcal{X}_{\mathcal{A}/\mathcal{G}} \mid success.$ Topology-related sufficient condition:  $C_{\mathcal{S}}(\mathcal{G}) \implies \forall X \text{ in } \mathcal{X}_{\mathcal{A}/\mathcal{G}}, success.$ 

- \* ロ \* \* @ \* \* 注 \* \* 注 \* のへで

#### Combination



An execution is an alternated sequence of relabellings and topological events:  $X = \mathcal{R}_{\mathcal{A}_{[t_{last-1}, t_{last}]}} \circ Event_{t_{last-1}} \circ .. \circ Event_{t_i} \circ \mathcal{R}_{\mathcal{A}_{[t_{i-1}, t_i]}} \circ .. \circ Event_{t_1} \circ \mathcal{R}_{\mathcal{A}_{[t_0, t_1]}}(\mathbf{G}_0)$ We note  $\mathcal{X}_{\mathcal{A}/\mathcal{G}}$  the set of all possible execution sequences of an algorithm  $\mathcal{A}$  over an evolving graph  $\mathcal{G}$ 

Topology-related necessary condition:  $\neg C_{\mathcal{N}}(\mathcal{G}) \Longrightarrow \nexists X \in \mathcal{X}_{\mathcal{A}/\mathcal{G}} \mid success.$ Topology-related sufficient condition:  $C_{\mathcal{S}}(\mathcal{G}) \Longrightarrow \forall X$  in  $\mathcal{X}_{\mathcal{A}/\mathcal{G}}$ , success.

 $\rightarrow$  possibility to prove formally that a given property is necessary, or sufficient.

э

イロト イポト イヨト イヨト

Introd	
000	

Applications

Conclusion

#### Classes of evolving graphs

#### Propagation algorithm conditions

In order to inform all the nodes, it is:

necessary that a journey exists from the emitter to every other node (C<sub>N</sub>).

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

Applications

#### Classes of evolving graphs

#### Propagation algorithm conditions

In order to inform all the nodes, it is:

- necessary that a journey exists from the emitter to every other node (C<sub>N</sub>).
- *sufficient* that a strict journey exists from the emitter to every other node  $(C_S)$ .

3

Applications

#### Classes of evolving graphs

#### Propagation algorithm conditions

In order to inform all the nodes, it is:

- necessary that a journey exists from the emitter to every other node (C<sub>N</sub>).
- *sufficient* that a strict journey exists from the emitter to every other node  $(C_S)$ .

### Resulting classes of evolving graphs (or dynamic networks)

▲ロト ▲圖 と ▲臣 と ▲臣 と 一臣 … のん

A.Casteigts, S.Chaumette, A.Ferreira

Applications

#### Classes of evolving graphs

#### Propagation algorithm conditions

In order to inform all the nodes, it is:

- necessary that a journey exists from the emitter to every other node (C<sub>N</sub>).
- *sufficient* that a strict journey exists from the emitter to every other node  $(C_S)$ .

### Resulting classes of evolving graphs (or dynamic networks)

 $\rightarrow$   $\mathcal{F}_1:$  graphs where  $\mathcal{C}_\mathcal{N}$  is verified for at least one node (1- $\mathcal{J}\text{-}*).$ 

3

<ロ> <同> <同> < 回> < 回>

Applications

#### Classes of evolving graphs

### Propagation algorithm conditions

In order to inform all the nodes, it is:

- necessary that a journey exists from the emitter to every other node (C<sub>N</sub>).
- *sufficient* that a strict journey exists from the emitter to every other node  $(C_S)$ .

### Resulting classes of evolving graphs (or dynamic networks)

- $\rightarrow$   $\mathcal{F}_1:$  graphs where  $\mathcal{C}_\mathcal{N}$  is verified for at least one node (1- $\mathcal{J}\text{-}*).$
- $\rightarrow$   $\mathcal{F}_3:$  same as  $\mathcal{F}_1$  but with strict journeys (1- $\mathcal{J}_{\textit{strict}}\text{-}*).$

3

イロン イロン イヨン イヨン

Applications

#### Classes of evolving graphs

#### Propagation algorithm conditions

In order to inform all the nodes, it is:

- necessary that a journey exists from the emitter to every other node (C<sub>N</sub>).
- *sufficient* that a strict journey exists from the emitter to every other node  $(C_S)$ .

### Resulting classes of evolving graphs (or dynamic networks)

- $\rightarrow$   $\mathcal{F}_1:$  graphs where  $\mathcal{C}_\mathcal{N}$  is verified for at least one node (1- $\mathcal{J}\text{-}*).$
- $\rightarrow$   $\mathcal{F}_3:$  same as  $\mathcal{F}_1$  but with strict journeys (1- $\mathcal{J}_{\textit{strict}}\text{-}*).$
- $\rightarrow \mathcal{F}_2:$  graphs where  $\mathcal{C}_\mathcal{S}$  is verified for all nodes (\*- $\mathcal{J}\text{-*}).$

3

イロト イポト イヨト イヨト

Applications

#### Classes of evolving graphs

### Propagation algorithm conditions

In order to inform all the nodes, it is:

- necessary that a journey exists from the emitter to every other node (C<sub>N</sub>).
- *sufficient* that a strict journey exists from the emitter to every other node  $(C_S)$ .

## Resulting classes of evolving graphs (or dynamic networks)

- $\rightarrow$   $\mathcal{F}_1:$  graphs where  $\mathcal{C}_\mathcal{N}$  is verified for at least one node (1- $\mathcal{J}\text{-}*).$
- $\rightarrow$   $\mathcal{F}_3:$  same as  $\mathcal{F}_1$  but with strict journeys (1- $\mathcal{J}_{\textit{strict}}\text{-}*).$
- $\rightarrow$   $\mathcal{F}_2:$  graphs where  $\mathcal{C}_\mathcal{S}$  is verified for all nodes (\*- $\mathcal{J}\text{-*}).$
- $\rightarrow$   $\mathcal{F}_4:$  same as  $\mathcal{F}_2$  but with strict journeys (\*- $\mathcal{J}_{\textit{strict}}\text{-*}).$

3

< 口 > < 回 > < 回 > < 回 > < 回 > .



Enumeration algorithm with a pre-selected counter

• initial states: (C, 1) for the counter, N for all other vertices.

• relabelling rule:  $\overset{C,i}{\bullet} \overset{N}{\bullet} \xrightarrow{} \overset{C,i+1}{\bullet} \overset{F}{\bullet}$ 

(N means non-counted, F means counted)

▲□▶ ▲□▶ ▲目▶ ▲目▶ = 目 - のへで

Introduction Purpose Applications Conclusion

Enumeration algorithm with a pre-selected counter

- initial states: (C, 1) for the counter, N for all other vertices.
- relabelling rule:  $\overset{C,i}{\bullet} \overset{N}{\longrightarrow} \overset{C,i+1}{\bullet} \overset{F}{\bullet}$

(N means non-counted, F means counted)

#### $\mathcal{C}_{\mathcal{N}},\,\mathcal{C}_{\mathcal{S}}$ and resulting classes

 C<sub>N</sub> = C<sub>S</sub>: the counter will share an edge with every other node (at possibly various times and durations).

Enumeration algorithm with a pre-selected counter

- initial states: (C, 1) for the counter, N for all other vertices.
- relabelling rule:  $\overset{C,i}{\bullet} \overset{N}{\bullet} \xrightarrow{\phantom{aaaa}} \overset{C,i+1}{\bullet} \overset{F}{\bullet}$

(N means non-counted, F means counted)

#### $\mathcal{C}_{\mathcal{N}},\,\mathcal{C}_{\mathcal{S}}$ and resulting classes

- C<sub>N</sub> = C<sub>S</sub>: the counter will share an edge with every other node (at possibly various times and durations).
  - $\rightarrow$   $\mathcal{F}_5$ : graphs where the condition holds for at least one vertex (also, failure whatever the counter if outside of this class).

Enumeration algorithm with a pre-selected counter

- initial states: (C, 1) for the counter, N for all other vertices.
- relabelling rule:  $\overset{C,i}{\bullet} \overset{N}{\bullet} \xrightarrow{\phantom{aaaa}} \overset{C,i+1}{\bullet} \overset{F}{\bullet}$

(N means non-counted, F means counted)

#### $\mathcal{C}_{\mathcal{N}},\,\mathcal{C}_{\mathcal{S}}$ and resulting classes

- C<sub>N</sub> = C<sub>S</sub>: the counter will share an edge with every other node (at possibly various times and durations).
  - $\rightarrow$   $\mathcal{F}_5$ : graphs where the condition holds for at least one vertex (also, failure whatever the counter if outside of this class).
  - $\rightarrow \mathcal{F}_6$ : graphs where the condition is verified for all nodes (success guaranteed whatever the counter. Also, not being in this class means that at least one node would fail as counter)

 Introduction
 Purpose
 Applications
 Conclusion

 000
 00
 00
 00
 00

 Classes of evolving graphs (3)

#### Decentralized counting algorithm

• initial states: (C, 1) for all vertices.

• relabelling rule: 
$$\overset{C,i}{\bullet} \overset{C,j}{\bullet} \overset{C,j}{\longrightarrow} \overset{C,i+j}{\bullet} \overset{F}{\bullet}$$

A.Casteigts, S.Chaumette, A.Ferreira

 Introduction
 Purpose
 Applications
 Conclusi

 coo
 coo
 coo
 coo
 coo

 Classes of evolving graphs (3)
 coo
 coo
 coo

#### Decentralized counting algorithm

- initial states: (C, 1) for all vertices.
- relabelling rule:  $\overset{C,i}{\bullet} \overset{C,j}{\bullet} \overset{C,j}{\longrightarrow} \overset{C,i+j}{\bullet} \overset{F}{\bullet}$

#### $\mathcal{C}_{\mathcal{N}}$ and resulting class

- $\blacksquare$   $\mathcal{C}_{\mathcal{N}}:$  at least one node can be reached by all the others by a journey.
  - $\rightarrow \mathcal{F}_7$ : idem.

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

▲□▶ ▲□▶ ▲目▶ ▲目▶ = 目 - のへで

Introduction 000	Purpose 00	Applications	Conclusion
Classification of dyna	mic networks		
	1-*	$1-\mathcal{J}_{strict}-*$	$1-\mathcal{J}-*$
	5	J-3	$\mathcal{F}_1$
$\mathcal{F}_{6}$	$\mathcal{F}_4$	$\mathcal{F}_2$	$\mathcal{F}_7$
* - *	*-Jstrict <sup>-</sup> *	*-J-*	*-7-1

◆□▶ ◆□▶ ◆目▶ ◆目▶ ◆□ ◆ ○ ◆

A.Casteigts, S.Chaumette, A.Ferreira

	Purpose	Applications	Conclusio
		0000000	
<u>CI : ()</u>	<u> </u>		

#### Classification of dynamic networks



▲□▶ ▲□▶ ▲臣▶ ▲臣▶ 三臣 - のへで

A.Casteigts, S.Chaumette, A.Ferreira

	Purpose	Applications
000	00	000000

### Classification of dynamic networks



▲□▶ ▲□▶ ▲目▶ ▲目▶ 目 のへで

A.Casteigts, S.Chaumette, A.Ferreira

Purpose

Conclusion

#### Classification of dynamic networks



◆□ > ◆□ > ◆臣 > ◆臣 > ─臣 ─ のへで

A.Casteigts, S.Chaumette, A.Ferreira

Purpose	Applications
	0000000

Conclusion

### Classification of dynamic networks



A.Casteigts, S.Chaumette, A.Ferreira

Introduction 000		Purpose 00	Applications ○○○○●○○	
A 1	<u> </u>			

#### Algorithms Comparison



A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

= 900

・ロン ・回 と ・ ヨ ・ ・ ヨ ・ ・

Introduction	Purpose	Applications
000	00	○○○○●○○



◆□ > ◆□ > ◆臣 > ◆臣 > ─臣 ─のへで

A.Casteigts, S.Chaumette, A.Ferreira

Introduction 000	Purpose 00	Applications ○○○●○○	Conclusion
Algorithms Compariso	on		
~	$\mathcal{F}_5  \mathcal{F}_3$	$ \mathcal{F}_1$	
	$\rightarrow$		
		< /	



It exists topologies where counting<sub>v1</sub> necessarily fails, while counting<sub>v2</sub> might have some chances of success.

-

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

Introduction 000	Purpose 00	Applications	Conclusion
Algorithms Con	nparison		
	$\mathcal{F}_5$	$ \mathcal{F}_3  \cdots  .$	$\mathcal{F}_1$
	$/ \rightarrow$	< $/$	
		$\sim$	
<i>F</i> <sub>6</sub>	$\longrightarrow \mathcal{F}_{A}$	$\rightarrow \mathcal{F}_2$	$\mathcal{F}_{7}$

It exists topologies where counting<sub>v1</sub> necessarily fails, while counting<sub>v2</sub> might have some chances of success.

 $C_{\mathcal{S}}(counting_{v1})$ 

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

2

・ロト ・回ト ・ヨト ・ヨト

Introduction 000	Purpose 00	Applications ○○○○●○○	Conclusion
Algorithms Cor	nparison		
	$\mathcal{F}_5$	$ \mathcal{F}_3  \mathcal{I}$	$F_1$
	$/ \rightarrow$	< $/$	
			-
$\mathcal{F}_6$ ———	$\longrightarrow \mathcal{F}_4$ —	$\longrightarrow \mathcal{F}_2 \longrightarrow \mathcal{F}_2$	F <sub>7</sub>

# $C_{S}(counting_{v1})$

- It exists topologies where counting<sub>v1</sub> necessarily fails, while counting<sub>v2</sub> might have some chances of success.
- But if we know that the condition of *counting<sub>v1</sub>* is matched, then better using this one.

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

A (1) > A (1) > A



- It exists topologies where counting<sub>v1</sub> necessarily fails, while counting<sub>v2</sub> might have some chances of success.
- But if we know that the condition of *counting*<sub>v1</sub> is matched, then better using this one.
- The choice depends on the expected properties of the target context.

э



- It exists topologies where counting<sub>v1</sub> necessarily fails, while counting<sub>v2</sub> might have some chances of success.
- But if we know that the condition of *counting*<sub>v1</sub> is matched, then better using this one.
- The choice depends on the expected properties of the target context.
- It would be interesting to now what properties the target context is likely to match.

イロト イポト イヨト イヨト

Introduction	Purpose	Applications	Conclusion
000	00	○○○○●●	
Automated Verification	on .		

$$\mathsf{Mobility}\;\mathsf{Model}\;{\longrightarrow}\;\mathsf{Generation}{\longrightarrow}\;\mathsf{Network}\;\mathsf{Traces}$$

・ロト ・団ト ・ヨト ・ヨー うべの

A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000	00	○○○○●○	
Automated Verification			



◆□ > ◆□ > ◆臣 > ◆臣 > 善臣 の Q @

A.Casteigts, S.Chaumette, A.Ferreira

	Purpose	Applications	Conclusion		
		0000000			
Automated Verification					



▲□▶ ▲□▶ ▲目▶ ▲目▶ ▲□ ▼ つへで

A.Casteigts, S.Chaumette, A.Ferreira

Introduction 000	Purpose 00	Applications ○○○○●○	Conclusion
Automated Ve	erification		
Algorithm ————————————————————————————————————	$\overbrace{\text{Analysis}} \longrightarrow \text{Conditions} \longrightarrow$	<del>````````````````````````````````</del>	
Mobility Model $\rightarrow$	$\overbrace{Generation} \rightarrow Network Traces$	s 🦳 Evolving Graph	
Real Network	Sensing → Network Traces	s 🦯 Instances	

Introduction 000	Purpose 00	Applications ○○○○●○	Conclusion
Automated Verification	n		
Algorithm ————————————————————————————————————	$\longrightarrow$ Conditions $\longrightarrow$	Evolving Graph Classes	
Mobility Model $\rightarrow$ Generation Real Network $\rightarrow$ Sensing)	$\stackrel{)\longrightarrow}{\longrightarrow} \text{Network Traces} $	Evolving Graph Instances	

A.Casteigts, S.Chaumette, A.Ferreira





 $\mathcal{C}_{\mathcal{N}}$  (or  $\mathcal{C}_{\mathcal{S}}$ ) is matched in a all/none/some cases?  $\implies$  decision.

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

3 x 3

Image: A math the second se
Introduction 000	Purpose 00	Applications ○○○○○●	Conclusion
Automated Ve	rification		
a• c 3,4•	e		
2	$\longrightarrow$		
$b \stackrel{\frown}{\longrightarrow} d$			
$\mathcal{G} = (\mathcal{G}, \mathcal{S}_{\mathcal{G}}, \mathcal{S}_{\mathbb{T}})$	)		

◆□> ◆□> ◆目> ◆目> ◆日> ● ● ●

A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
000	00	○○○○○●	
Automated Verification	on		





 $\mathcal{G} = (\mathcal{G}, \mathcal{S}_{\mathcal{G}}, \mathcal{S}_{\mathbb{T}})$ 

(Underlying graph)

▲ロト ▲御 ト ▲臣 ト ▲臣 ト ― 臣 ― 釣んで

A.Casteigts, S.Chaumette, A.Ferreira

Introduction	Purpose	Applications	Conclusion
Automated Verification	on		
a 1 _ 3.4 ∕•e		C A LA	
	a c e	a. e	
$b \bullet \overbrace{2}^{2} \bullet d \longrightarrow$	b d	b d	
-	G	Н	
$\mathcal{G} = (\mathcal{G}, \mathcal{S}_{\mathcal{G}}, \mathcal{S}_{\mathbb{T}})$	(Underlying graph)	(Transitive closure of	
		journeys)	

Introduction 000	Purpose 00	Applications ○○○○○○●		Conclusion
Automated Verificatio	n			
$\begin{array}{c} a \bullet & I & c & 3, \bullet & \bullet \\ & & & & \\ & & & & \\ b \bullet & & & 2 \\ & & & \\ \mathcal{G} = (G, \mathcal{S}_G, \mathcal{S}_{\mathbb{T}}) \end{array}$	$a \cdot c \cdot e \\ b \cdot d \\ G \\ (Underlying graph)$	a b b H (Transitive closure of journeys)	<i>c</i> <i>b</i> <i>d</i> <i>H</i> <sub>strict</sub> (Transitive closure o strict journeys)	f

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

5 DQC

・ロン ・回 と ・ ヨ ・ ・ ヨ ・ ・

Introduction 000	Purpose 00	Applications	Conclusion
Automated Verification	on		
$\begin{array}{c} a \bullet & f \\ & & e \\ & & & e \\ & & & & e \\ & & & &$	$a \cdot c \cdot e$ b \cdot d G (Underlying graph)	e b H (Transitive closure of journeys)	<i>c</i> <i>e</i> <i>b</i> <i>d</i> <i>H</i> <i>strict</i> (Transitive closure of strict journeys)
$\begin{array}{lll} \mathcal{G} \in \mathcal{F}_1 & (1\text{-}\mathcal{J}\text{-}*) \\ \mathcal{G} \in \mathcal{F}_2 & (*\text{-}\mathcal{J}\text{-}*) \\ \mathcal{G} \in \mathcal{F}_3 & (1\text{-}\mathcal{J}\text{strict}\text{-}*) \\ \mathcal{G} \in \mathcal{F}_4 & (*\text{-}\mathcal{J}\text{strict}\text{-}*) \\ \mathcal{G} \in \mathcal{F}_5 & (1-*) \\ \mathcal{G} \in \mathcal{F}_6 & (*-*) \\ \mathcal{G} \in \mathcal{F}_7 & (*\text{-}\mathcal{J}\text{-}1) \end{array}$	$\begin{array}{ccc} \Longleftrightarrow & H \text{ conta} \\ \Leftrightarrow & H \text{ is a co} \\ \Leftrightarrow & H_{strict} \text{ co} \\ \Leftrightarrow & H_{strict} \text{ is} \\ \Leftrightarrow & G \text{ conta} \\ \Leftrightarrow & G \text{ is a co} \\ \Leftrightarrow & H \text{ conta} \end{array}$	ins an out-dominating omplete graph. ontains an out-domin. a complete graph. ins a dominating set omplete graph. ins an in-dominating	g set of size 1. ating set of size 1. of size 1. set of size 1.

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

三 のへで

・ロン ・四 と ・ ヨ と ・ ヨ と

Introduction 000	Purpose 00	Applications 0000000	Conclusion

 Undirected graphs, bandwidth limitations, latency. (not restricted by the models)

Introd	

- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.



Introduction	Purpose	Applications	Conclusion
000	00	0000000	

- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.



Introduction	Purpose	Applications	Conclusion
000	00	0000000	

- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.



- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.
- Scale to more complex algorithms?

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.
- Scale to more complex algorithms?

#### Prospects

Introd	

- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.
- Scale to more complex algorithms?

#### Prospects

new algorithms to be characterized

- ▲日 > ▲ 国 > ▲ 国 > ▲ 国 > ▲ 日 >

A.Casteigts, S.Chaumette, A.Ferreira

- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.
- Scale to more complex algorithms?

## Prospects

- new algorithms to be characterized
- new resulting classes of evolving graphs

A.Casteigts, S.Chaumette, A.Ferreira

Characterizing Topological Assumptions of Dist. Algo. in Dynamic Networks

э

(日) (同) (三) (三)

- Undirected graphs, bandwidth limitations, latency. (not restricted by the models)
- Topology-related conditions.
- Scale to more complex algorithms?

#### Prospects

- new algorithms to be characterized
- new resulting classes of evolving graphs
- some insights about the networking impact of mobility

э

(日) (同) (三) (三)

Introduction	Purpose 00	Applications	Conclusion

# Thank you

# Questions?

▲ロ → ▲ □ → ▲ □ → ▲ □ → ▲ □ → ▲ □ → ▲ □ → ▲ □ → ▲ □ → ▲ □ →

A.Casteigts, S.Chaumette, A.Ferreira