Parameterized Verification of Synchronization in constrained reconfigurable broadcast networks

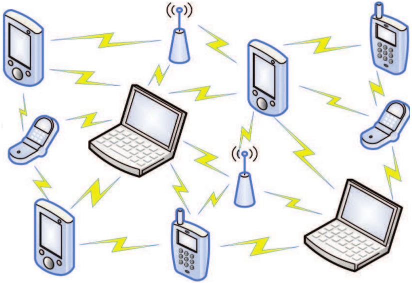
Nathalie Bertrand

Inria & IRISA

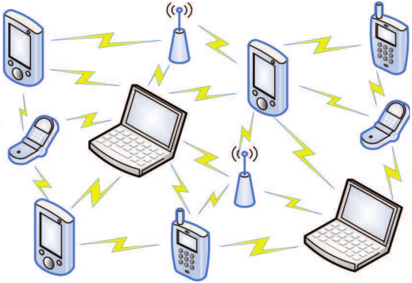
**joint work with Balasubramanian A.R. (CMI)
and Nicolas Markey (CNRS & IRISA)**

TACAS @ ETAPS 2018

Formal Models for Ad Hoc Networks



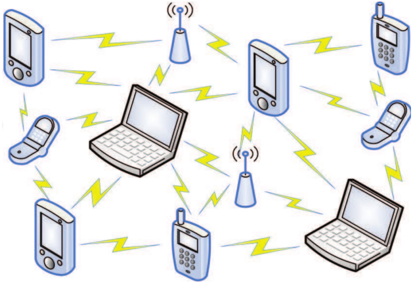
Formal Models for Ad Hoc Networks



important features

- mobile nodes
- unknown initial topology
- communication by broadcasts to neighbours

Formal Models for Ad Hoc Networks



important features

- mobile nodes
- unknown initial topology
- communication by broadcasts to neighbours

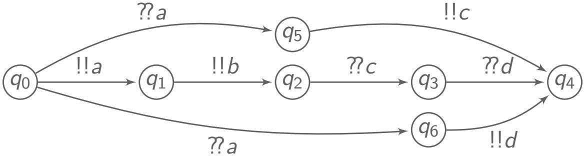
broadcast networks with evolving communication topology

Reconfigurable Broadcast Networks

- nodes execute each a finite-state process
- finitely many types of processes
- broadcasts to neighbours
- evolving communication topology

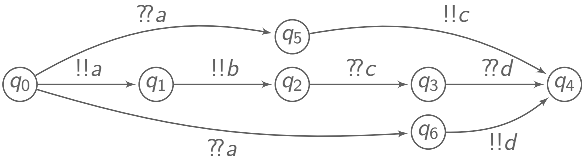
Reconfigurable Broadcast Networks

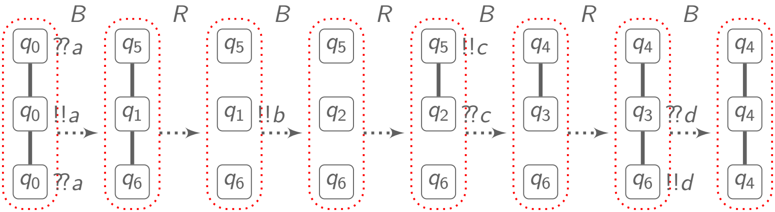
- nodes execute each a finite-state process
- finitely many types of processes
- broadcasts to neighbours
- evolving communication topology



Reconfigurable Broadcast Networks

- nodes execute each a finite-state process
- finitely many types of processes
- broadcasts to neighbours
- evolving communication topology





Parameterized Verification of Broadcast Networks

parameter: initial configuration (number of nodes and topology)

Parameterized verification: check that a property holds for all parameter values, and all executions.

Parameterized Verification of Broadcast Networks

parameter: initial configuration (number of nodes and topology)

Parameterized verification: check that a property holds for all parameter values, and all executions.

Dually, looking for counterexamples: $\exists \gamma_0 \to \gamma_1 \to \cdots \models \varphi$?

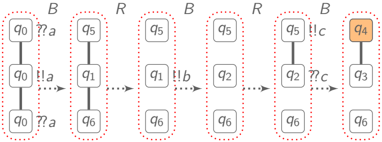
Parameterized Verification of Broadcast Networks

parameter: initial configuration (number of nodes and topology)

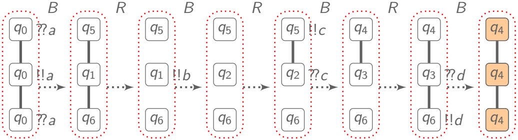
Parameterized verification: check that a property holds for all parameter values, and all executions.

Dually, looking for counterexamples: $\exists \gamma_0 \to \gamma_1 \to \cdots \models \varphi$?

- Reachability



- Synchronization



Parameterized Verification of Broadcast Networks

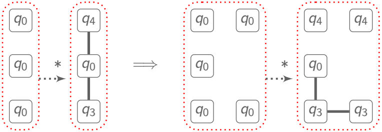
Theorem [Delzanno et al. Concur'10]
Parameterized reachability and synchronization are decidable in PTIME.

Parameterized Verification of Broadcast Networks

Theorem [Delzanno et al. Concur'10]
Parameterized reachability and synchronization are decidable in PTIME.

- reachability: saturation algorithm, starting with $\{q_0\}$
- synchronization: forward and backward iterative pruning of broadcast protocol, using algorithm for reachability

Monotonicity property: if a configuration is reachable, then all configurations obtained by duplicating some nodes are also reachable

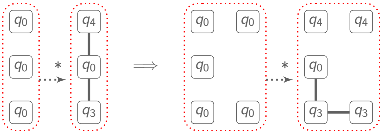


Parameterized Verification of Broadcast Networks

Theorem [Delzanno et al. Concur'10]
Parameterized reachability and synchronization are decidable in PTIME.

- reachability: saturation algorithm, starting with $\{q_0\}$
- synchronization: forward and backward iterative pruning of broadcast protocol, using algorithm for reachability

Monotonicity property: if a configuration is reachable, then all configurations obtained by duplicating some nodes are also reachable



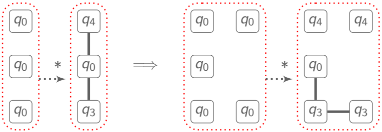
Rk: both problems are undecidable under fixed topology

Parameterized Verification of Broadcast Networks

Theorem [Delzanno et al. Concur'10]
Parameterized reachability and synchronization are decidable in PTIME.

- reachability: saturation algorithm, starting with $\{q_0\}$
- synchronization: forward and backward iterative pruning of broadcast protocol, using algorithm for reachability

Monotonicity property: if a configuration is reachable, then all configurations obtained by duplicating some nodes are also reachable



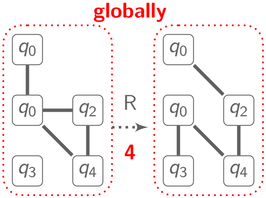
Rk: both problems are undecidable under fixed topology

Limitation: arbitrary reconfigurations between communications

this talk: towards realistic representation of mobility

Constraints on reconfiguration

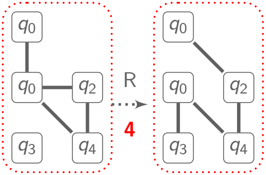
Quantifying reconfiguration: number of modified links



Constraints on reconfiguration

Quantifying reconfiguration: number of modified links

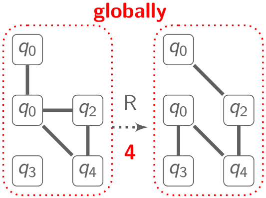
globally



k-bounded execution

Constraints on reconfiguration

Quantifying reconfiguration: number of modified links



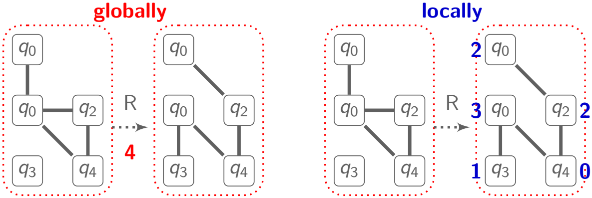
k-bounded execution

k-bounded-on-average execution

$\frac{1}{m} \sum_{i=1}^{m} \mathbf{k_i} \leq \mathbf{k}$

Constraints on reconfiguration

Quantifying reconfiguration: number of modified links



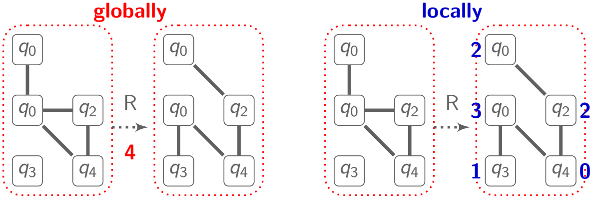
k-bounded execution

k-bounded-on-average execution

$\frac{1}{m}\sum_{i=1}^{m} \mathbf{k_i} \leq \mathbf{k}$

Constraints on reconfiguration

Quantifying reconfiguration: number of modified links



k-bounded execution

k-bounded-on-average execution

$\frac{1}{m}\sum_{i=1}^{m} \mathbf{k_i} \leq \mathbf{k}$

locally-*k*-bounded execution

Reachability under reconfiguration constraints

There exists a reaching execution iff
there exists a $\star$-bounded reaching execution.

Proof idea: interleave several copies of a witness execution
only the first copy needs to reach the target

Reachability under reconfiguration constraints

There exists a reaching execution iff
there exists a $\star$-bounded reaching execution.

Proof idea: interleave several copies of a witness execution
only the first copy needs to reach the target

Rest of the talk: synchronization

Classification of constraints

There exists a k-bounded synchronizing execution iff
there exists a k-bounded-on-average synchronizing execution.

Classification of constraints

There exists a k-bounded synchronizing execution iff
there exists a k-bounded-on-average synchronizing execution.

Proof idea for right-to-left implication with $k = 1$
Potential function along execution

- initially 0
- increases by k (here 1) in broadcast steps
- decreases by number of modified links in reconfigurations steps

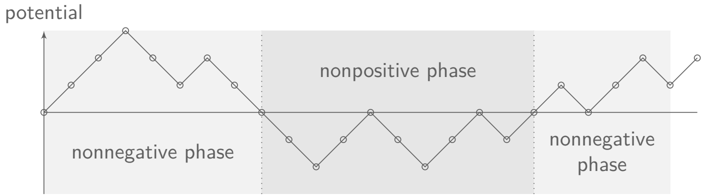
Classification of constraints

There exists a k-bounded synchronizing execution iff
there exists a k-bounded-on-average synchronizing execution.

Proof idea for right-to-left implication with $k = 1$
Potential function along execution

- initially 0
- increases by k (here 1) in broadcast steps
- decreases by number of modified links in reconfigurations steps

Decomposition of execution w.r.t sign of potential



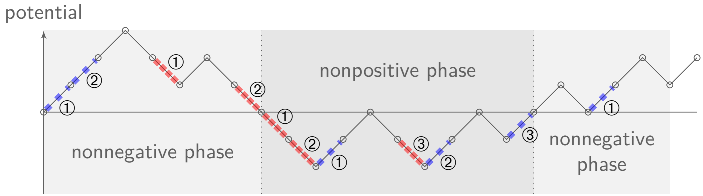
Classification of constraints

There exists a k-bounded synchronizing execution iff
there exists a k-bounded-on-average synchronizing execution.

Proof idea for right-to-left implication with $k = 1$
Potential function along execution

- initially 0
- increases by k (here 1) in broadcast steps
- decreases by number of modified links in reconfigurations steps

Decomposition of execution w.r.t sign of potential



Correspondence between repeated broadcasts, and repeated link modifications

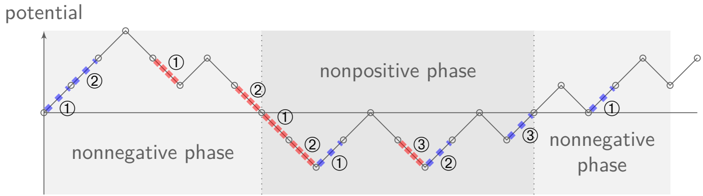
Classification of constraints

There exists a k-bounded synchronizing execution iff
there exists a k-bounded-on-average synchronizing execution.

Proof idea for right-to-left implication with $k = 1$
Potential function along execution

- initially 0
- increases by k (here 1) in broadcast steps
- decreases by number of modified links in reconfigurations steps

Decomposition of execution w.r.t sign of potential



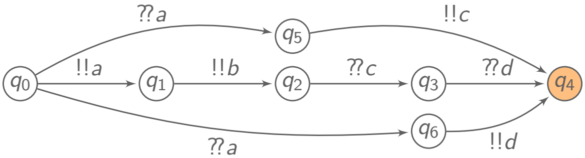
Correspondence between repeated broadcasts, and repeated link modifications
Interleave duplicates of initial execution to strictly alternate $+1$ and -1 steps

Classification of constraints

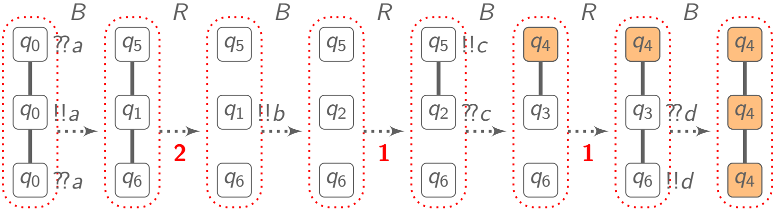
There may exist a synchronizing execution, while no 1-bounded synchronizing execution exists.

Classification of constraints

There may exist a synchronizing execution, while no 1-bounded synchronizing execution exists.

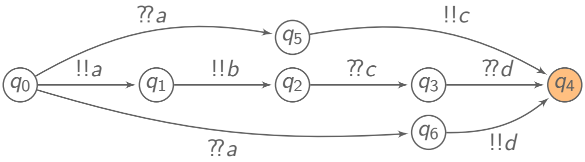


unspecified receptions lead to a sink



Classification of constraints

There may exist a synchronizing execution, while no 1-bounded synchronizing execution exists.



unspecified receptions lead to a sink

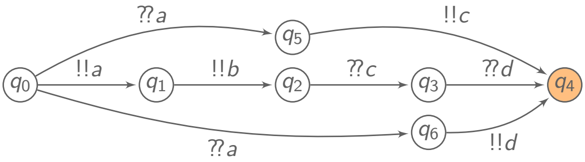
♯ nodes: x for top branch; y for central branch; z: for bottom branch

In any synchronizing execution

- ♯ broadcast steps $= x + 2y + z$
- ♯ reconfigured links $\geq x + 2y + z$

Classification of constraints

There may exist a synchronizing execution, while no 1-bounded synchronizing execution exists.



unspecified receptions lead to a sink

♯ nodes: x for top branch; y for central branch; z: for bottom branch

In any synchronizing execution

- ♯ broadcast steps $= x + 2y + z$
- ♯ reconfigured links $\geq x + 2y + z$

In 1-bounded executions: ♯ broadcast steps $\geq 1+$♯ reconfigured links

Classification of constraints

There exists a synchronizing execution iff
there exists a 1-locally-bounded synchronizing execution.

Classification of constraints

There exists a synchronizing execution iff
there exists a 1-locally-bounded synchronizing execution.

For $f : \mathbb{N} \to \mathbb{N}$ is a diverging function
$f(n)$-bounded: reconfigurations in executions over configurations of size n concern at most $f(n)$ link

There exists a synchronizing execution iff
there exists a $f(n)$-bounded synchronizing execution.

Classification of constraints

There exists a synchronizing execution iff
there exists a 1-locally-bounded synchronizing execution.

For $f : \mathbb{N} \rightarrow \mathbb{N}$ is a diverging function
$f(n)$-bounded: reconfigurations in executions over configurations of size n concern at most $f(n)$ link

There exists a synchronizing execution iff
there exists a $f(n)$-bounded synchronizing execution.

Corollary
The parameterized synchronization problem is in PTIME when restricting to either 1-locally-bounded, or $f(n)$-bounded executions.

Focus on k-constrained synchronization

The parameterized synchronization problem is undecidable when restricting to k-**bounded executions**.
It is decidable for k-**bounded executions** with **rendez-vous**.

Focus on k-constrained synchronization

The parameterized synchronization problem is undecidable when restricting to k-**bounded executions**.
It is decidable for k-**bounded executions** with **rendez-vous**.

Proof idea: encoding into Petri net reachability

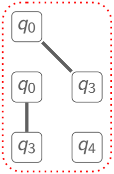
rendez-vous communications
$\Rightarrow$ at most one neighbour per node
finitely many patterns

Focus on k-constrained synchronization

The parameterized synchronization problem is undecidable when restricting to k-**bounded executions**.
It is decidable for k-**bounded executions** with **rendez-vous**.

Proof idea: encoding into Petri net reachability

rendez-vous communications
$\Rightarrow$ at most one neighbour per node
finitely many patterns

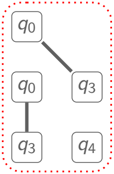


$q_4 + (q_0, q_3) + (q_0, q_3)$

Focus on k-constrained synchronization

The parameterized synchronization problem is undecidable when restricting to k-**bounded executions**.
It is decidable for k-**bounded executions** with **rendez-vous**.

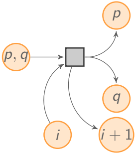
Proof idea: encoding into Petri net reachability

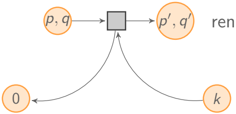


$q_4 + (q_0, q_3) + (q_0, q_3)$

rendez-vous communications
⇒ at most one neighbour per node
finitely many patterns

reconfiguration



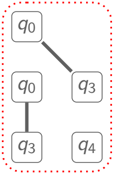


rendez-vous

Focus on k-constrained synchronization

The parameterized synchronization problem is undecidable when restricting to k-**bounded executions**.
It is decidable for k-**bounded executions** with **rendez-vous**.

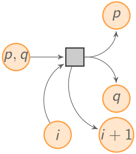
Proof idea: encoding into Petri net reachability

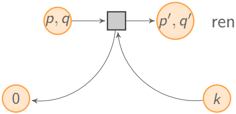


$q_4 + (q_0, q_3) + (q_0, q_3)$

rendez-vous communications
$\Rightarrow$ at most one neighbour per node
finitely many patterns

reconfiguration





rendez-vous

initialisation generate tokens in places q_0 and (q_0, q_0)

simulation reconfigurations and rendez-vous communications

termination emptying tokens from target place

Summary of results

Constraining reconfigurations in broadcast protocols

Contributions

- equivalence of k-bounded and k-bounded-on-average
- non-equivalence of 1-bounded with unconstrained reconfiguration
- undecidability of synchronization under k-bounded assumption
- decidability assuming rendez-vous communications

Techniques

- duplication of executions and clever interleaving
- encoding of 2-counter machine
- reduction to reachability in Petri net

Future work

Quantitative impact of reconfiguration restrictions

- minimum number of nodes in a witness execution
- mininum number of steps to synchronize

Long-term objective: realistic model of mobility in ad hoc networks

- representing with physical constraints: speed of nodes vs communication rate
- exploiting statistical analysis of real behaviours