Note that the scheduler uses the processor

if it successfully, it later uses

we move to a location where no time can pass and immediate notify the scheduler of either the success or failure of the computation. The automaton for the scheduler also changes for this model since it must

Adding a Faulty Processor.

In this paper, we have presented an introduction to the model of probabilistic timed automata. Other important issues to investigate in the context of PTAs include robustness [7].

Acknowledgments.

Tomata. Other important issues to investigate in the context of PTAs include robustness [7].
PRISM-games 3.0

Stochastic Game Verification
with
Concurrency, Equilibria and Time

Marta Kwiatkowska, Gethin Norman, Dave Parker, Gabriel Santos
Stochastic Games

• Stochastic games:
  – **nondeterminism**: adversarial, controllers, …
  – **probability**: randomisation, failures, noise, …
  – **players**: competing/collaborating components
  – **strategies**: rational decisions made by players
  – **costs** (resources) & **rewards** (incentives)

• Applications:
  – computer security, network/communication protocols, algorithms for distributed consensus, energy management, autonomous robotics, auctions, …

• Challenge:
  – how to design these systems correctly?
  – complex interactions between features
PRISM–games

- PRISM–games
  - modelling and analysis of stochastic games
  - automated verification or synthesis of strategies with quantitative guarantees

- Example specification in rPATL
  - $\langle \langle \text{robot}_1 \rangle \rangle P_{\geq 0.95} [ F_{\leq 10} \text{goal}_1 ]$
  - “robot 1 has a strategy to ensure that, with probability at least 0.95, it reaches its goal in 10 steps, regardless of the strategies of other robots”
Concurrent stochastic games

Turn-based stochastic games (TSGs)

Concurrent stochastic games (CSGs)

- Players make concurrent, independent decisions
Concurrent stochastic games

Example: Maximising expected investor profit in a futures market

- Players make concurrent, independent decisions

Too pessimistic: unrealistic strategy for adversary
CSCG in PRISM—games 3.0

csg
player p1 user1 endplayer
player p2 user2 endplayer
// Users (senders)
mODULE user1
  s1 : [0..1] init 0; // has player 1 sent?
  e1 : [0..emax] init emax; // energy level of player 1
  w1 true -> (s1'=0); // wait
  t1 e1>0 -> (s1'=c' ? 0 : 1) & (e1'=e1-1); // transmit
ENDMODULE

MODULE user2 = user1 [ s1=s2, e1=e2, w1=w2, t1=t2 ] ENDMODULE
// Channel: used to compute joint probability distribution for transmission failure
mODULE channel
  c : bool init false; // is there a collision?
  t1 w1 true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 1 transmits
  t1 t2 true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 2 transmits
  t1 t2 true -> q2 : (c'=false) + (1-q2) : (c'=true); // both users transmit
ENDMODULE
CSGs in PRISM—games 3.0

player p1 user1 endplayer
player p2 user2 endplayer

// Users (senders)
module user1
    s1 : [0..1] init 0; // has player 1 sent?
    e1 : [0..emax] init emax; // energy level of player 1
    [w1] true -> (s1'=0); // wait
    [t1] e1>0 -> (s1'=c'?0:1) & (e1'=e1-1); // transmit
endmodule

module user2 = user1 [ s1=s2, e1=e2, w1=w2, t1=t2 ] endmodule

// Channel: used to compute joint probability distribution for transmission failure
module channel
    c : bool init false; // is there a collision?
    [t1,w2] true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 1 transmits
    [w1,t2] true -> q1 : (c'=false) + (1-q1) : (c'=true); // only user 2 transmits
    [t1,t2] true -> q2 : (c'=false) + (1-q2) : (c'=true); // both users transmit
endmodule
Equilibria-based properties

- **Example: multi-robot coordination**
  - $\langle\langle\text{robot}_1\rangle\rangle_{\text{max}=?} \ P [F \leq k \ \text{goal}_1]$
  - Maximise probability of robot 1 reaching its goal, regardless of robot 2
  - $\langle\langle\text{robot}_1: \text{robot}_2\rangle\rangle_{\text{max}=?} \ (P [F \leq k \ \text{goal}_1] + P [F \leq k \ \text{goal}_2])$
  - Find strategies where robots 1 & 2 have no incentive to change actions and maximise joint goal probability

Zero-sum properties

Social-welfare optimal Nash equilibrium
PRISM–games 3.0

- Probabilistic timed games (turn-based)

- 10 new/expanded case studies
  - multi-robot coordination, network trust models, Aloha, intrusion detection, public good games, ...

- More information at:
  - prismmodelchecker.org/games/
  - documentation, examples, case studies, papers
  - downloads: 🍎 🐧 💻 + CAV’20 artefact VM

- Open source (GPLv2)
  - github.com/prismmodelchecker/prism-games