Towards Inverse Uncertainty Quantification in Software Development

Matteo Camilli [1], Angelo Gargantini [2], Patrizia Scandurra [2], Carlo Bellettini [1]

[1] Dept. of Computer Science, Università degli Studi di Milano, Italy
[2] DIGIP, Università degli Studi di Bergamo, Italy

matteo.camilli@unimi.it
http://camilli.di.unimi.it

15th International conference on Software Engineering and Formal Methods
SEFM 2017

Outline

- Introduction & Background
  - Uncertainty
  - Bayesian reasoning
- Envisioned Approach
  - IUQ by combining Online MBT & Bayesian reasoning
- Current stage of the work
- Conclusion

Introduction

Introduction

- Uncertainty
  - imperfect and/or unknown information
  - $S_?, E_? \models R$ @ Designtime
  - sources of uncertainty (epistemic or aleatory) can be introduced into the system by
    - the system itself: structural uncertainty, algorithmic uncertainty, etc.
    - the environment: parameter uncertainty, usage profiles, network latency, etc.

Introduction

- Uncertainty
  - imperfect and/or unknown information
  - $S_?, E_? \models R$ @ Designtime
  - sources of uncertainty (epistemic or aleatory) can be introduced into the system by
    - the system itself: structural uncertainty, algorithmic uncertainty, etc.
    - the environment: parameter uncertainty, usage profiles, network latency, etc.
- Problem
  - mathematical models are often imperfect and measured data from a running system is subject to noise

Introduction

- Uncertainty
  - imperfect and/or unknown information
  - $S_?, E_? \models R$ @ Designtime
  - sources of uncertainty (epistemic or aleatory) can be introduced into the system by
    - the system itself: structural uncertainty, algorithmic uncertainty, etc.
    - the environment: parameter uncertainty, usage profiles, network latency, etc.
- Problem
  - mathematical models are often imperfect and measured data from a running system is subject to noise
- Objective
  - Inverse Uncertainty Quantification (IUQ) in software development
    - discrepancy between the measured data at runtime and the formal specification (i.e., a mathematical model)
    - value calibration of unknown parameters in the model

State of the art

State of the art

- FUQ problem
  - [Epifani, C. et al., 2009] Model evolution by run-time parameter adaptation, in 2009 IEEE 31st International Conference on Software Engineering, May 2009, pp. 111–121
  - [N. Esfahani et al., 2013] Uncertainty in Self-Adaptive Software Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 214–238
  - [Perez-Palacin et al., 2014] Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation, in Proceedings of the 5th ACM/SPEC Int. Conference on Performance Engineering, 2014, pp. 3–14
  - …

State of the art

- FUQ problem
    - [Epifani, C. et al., 2009] Model evolution by run-time parameter adaptation, in 2009 IEEE 31st International Conference on Software Engineering, May 2009, pp. 111–121
    - [N. Esfahani et al., 2013] Uncertainty in Self-Adaptive Software Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 214–238
    - [Perez-Palacin et al., 2014] Uncertainties in the modeling of self-adaptive systems: A taxonomy and an example of availability evaluation, in Proceedings of the 5th ACM/SPEC Int. Conference on Performance Engineering, 2014, pp. 3–14
    - …
- Testing and Uncertainty
    - [A. S. Namin et al., 2010] Bayesian reasoning for software testing, in Proc. of the FSE/SDP Workshop on Future of Soft. Eng. Research, 2010, pp. 349–354
    - [N. Walkinshaw et al., 2017] Uncertainty-driven black-box test data generation, in IEEE Int. Conf. on Software Testing, Verification and Validation, 2017
    - …

Background

Background

- Specification formalism
  - Markov models s.t. the **system state is fully observable** and outcomes are partially under the control of a **decision maker**
    - **CTMDPs** —> CTMCs extension, nicely supported by powerful off-the-shelf model checking tools (e.g. MRMC - Markov Reward Model Checker)

Background

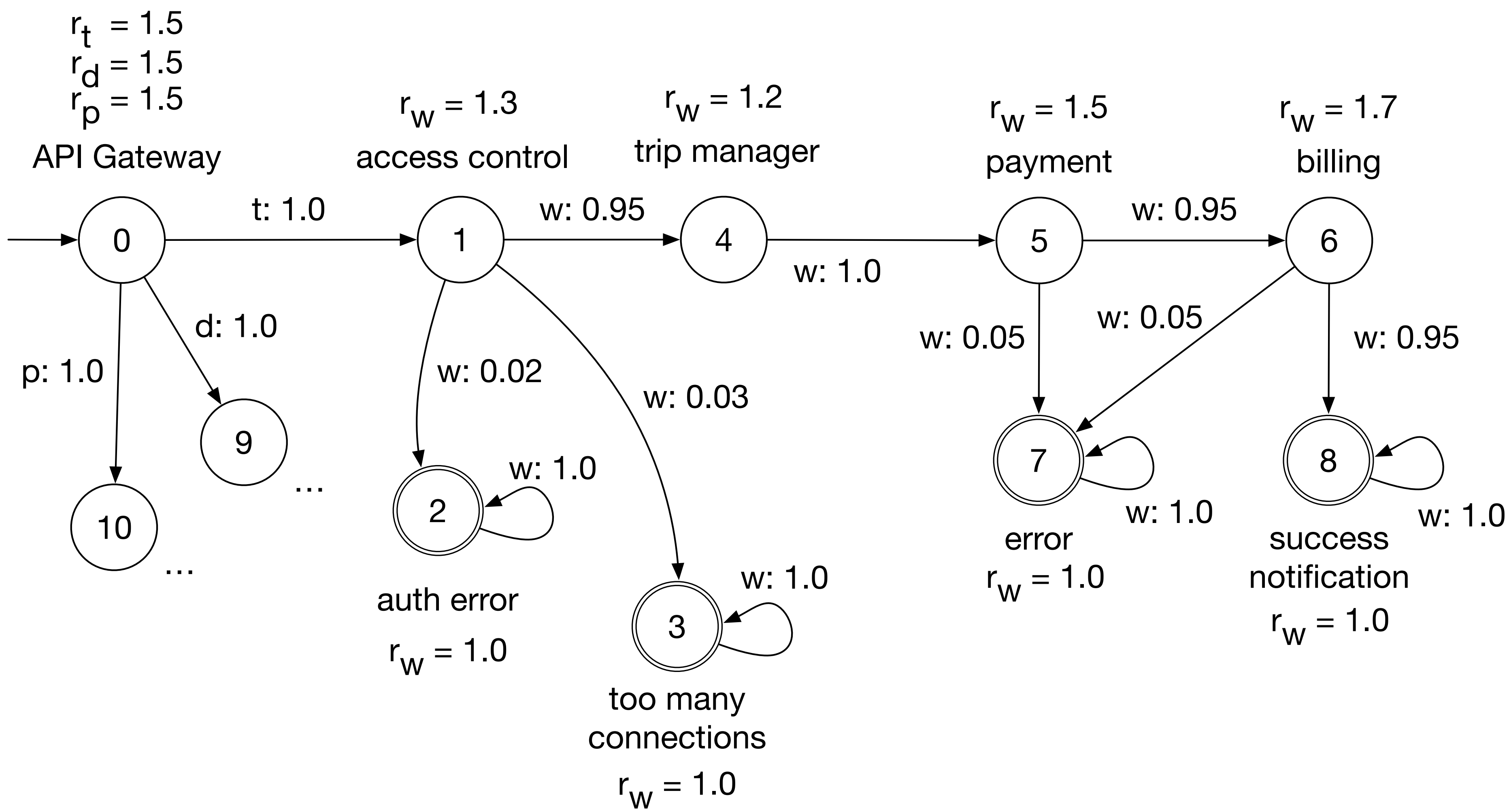
- Specification formalism
  - Markov models s.t. the **system state is fully observable** and outcomes are partially under the control of a **decision maker**
    - **CTMDPs** —> CTMCs extension, nicely supported by powerful off-the-shelf model checking tools (e.g. MRMC - Markov Reward Model Checker)
- Bayesian inference
  - $y^m(\Theta) \simeq y^e$
  - methodology: $P(\Theta \mid y^e) \propto P(y^e \mid \Theta) \cdot P(\Theta)$

Prior probability

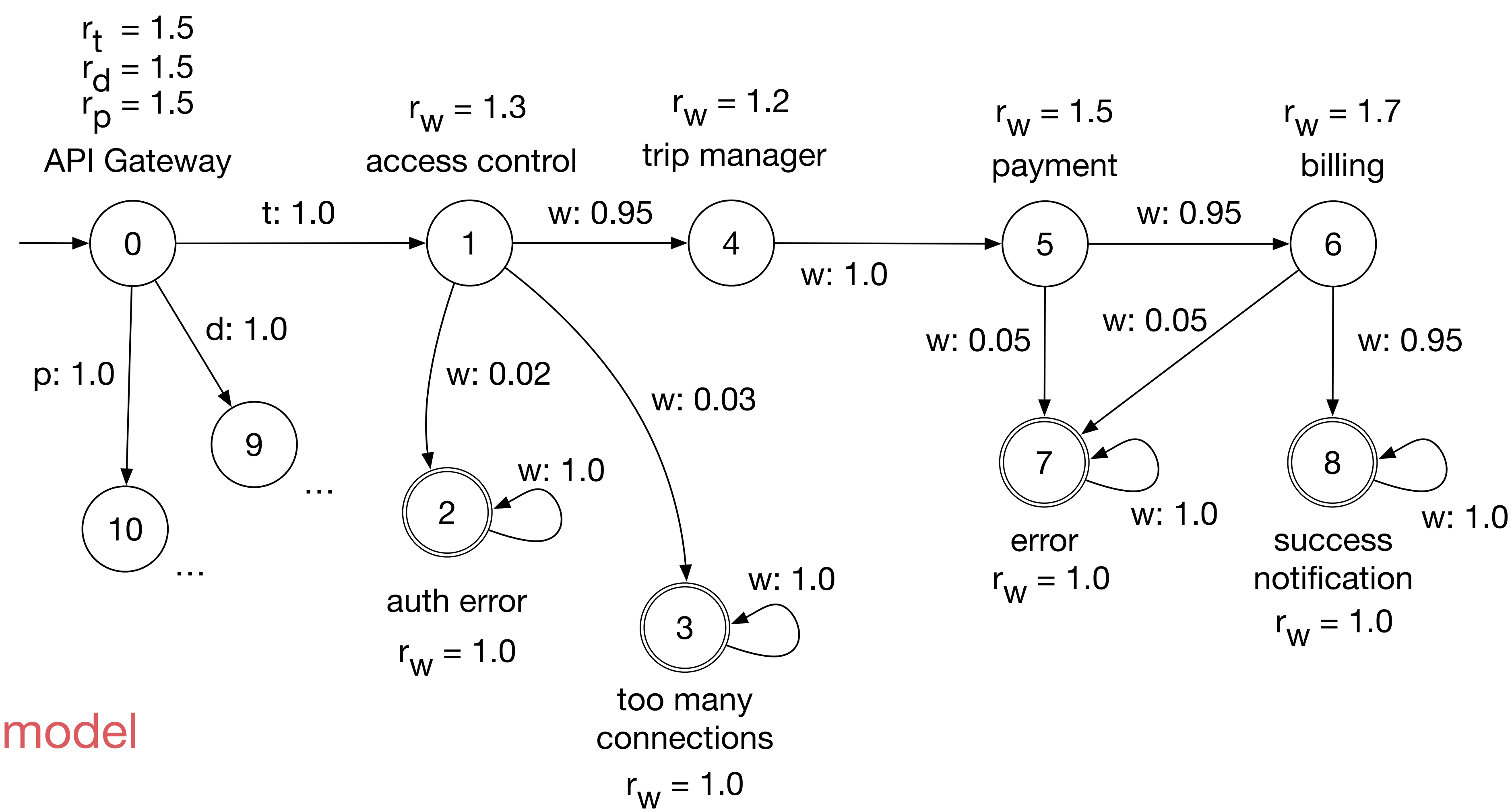
likelihood

Posterior probability

A running example: the taxi-hailing application

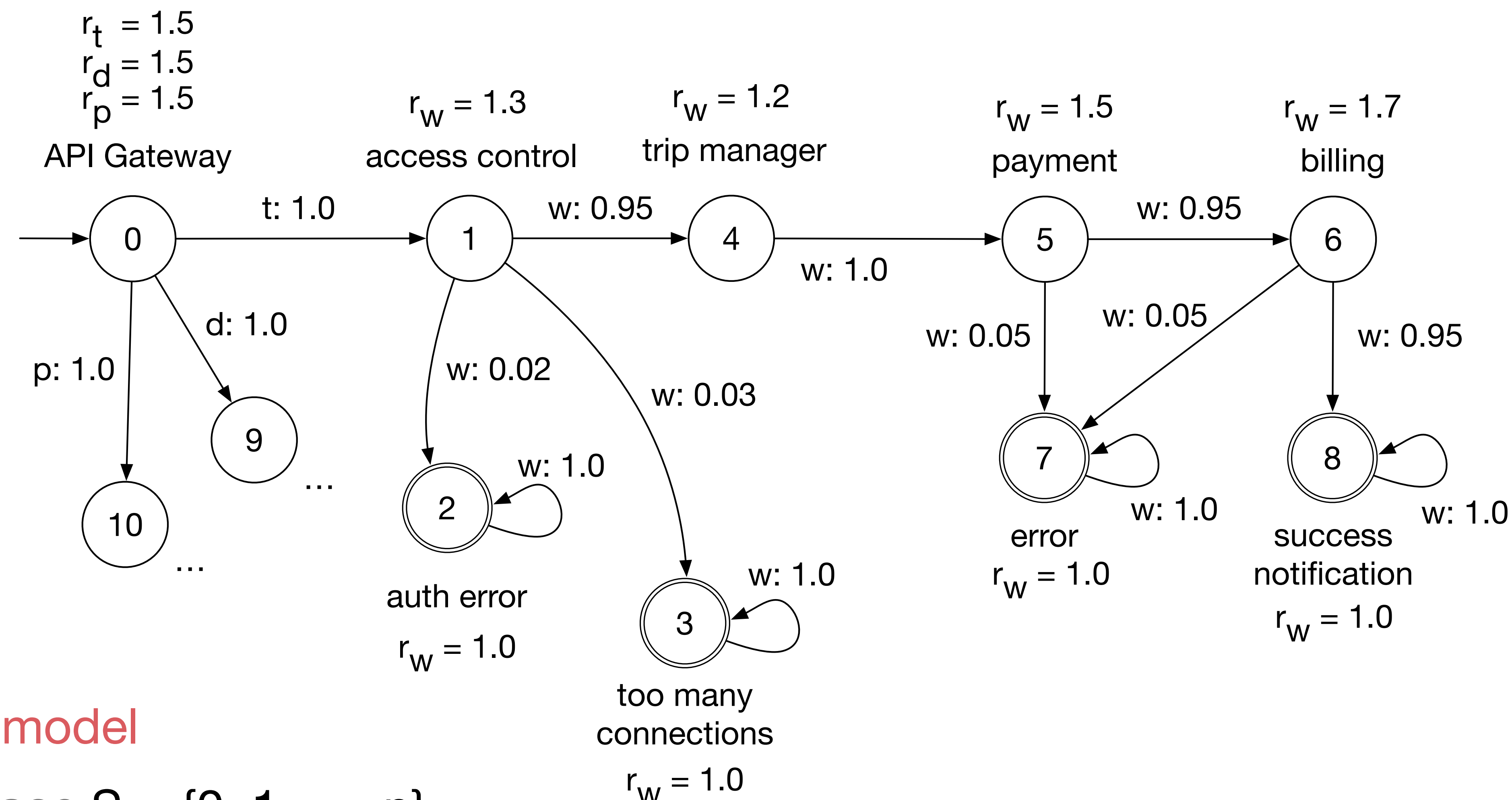


A running example: the taxi-hailing application



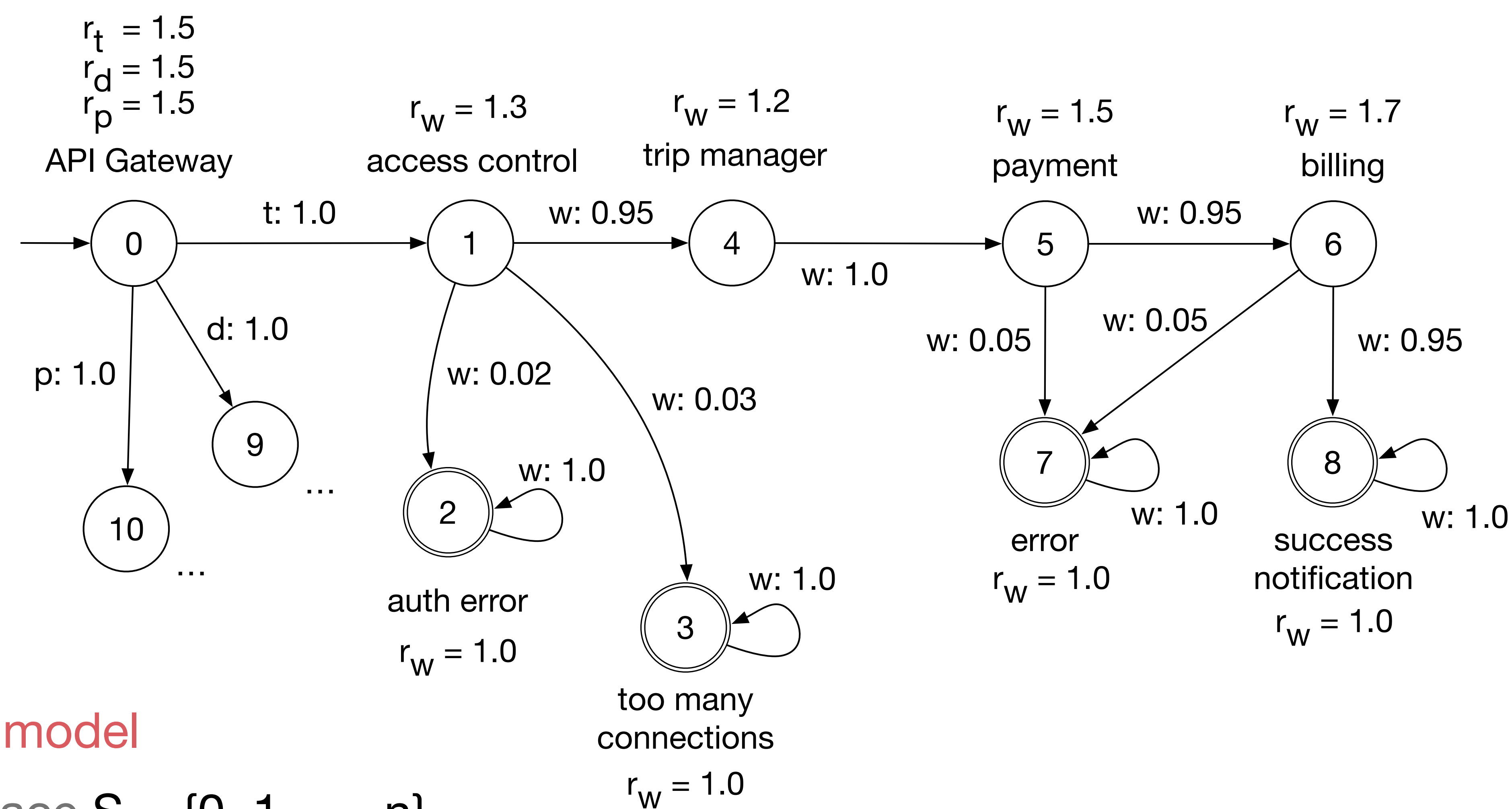
- CTMDP model

A running example: the taxi-hailing application



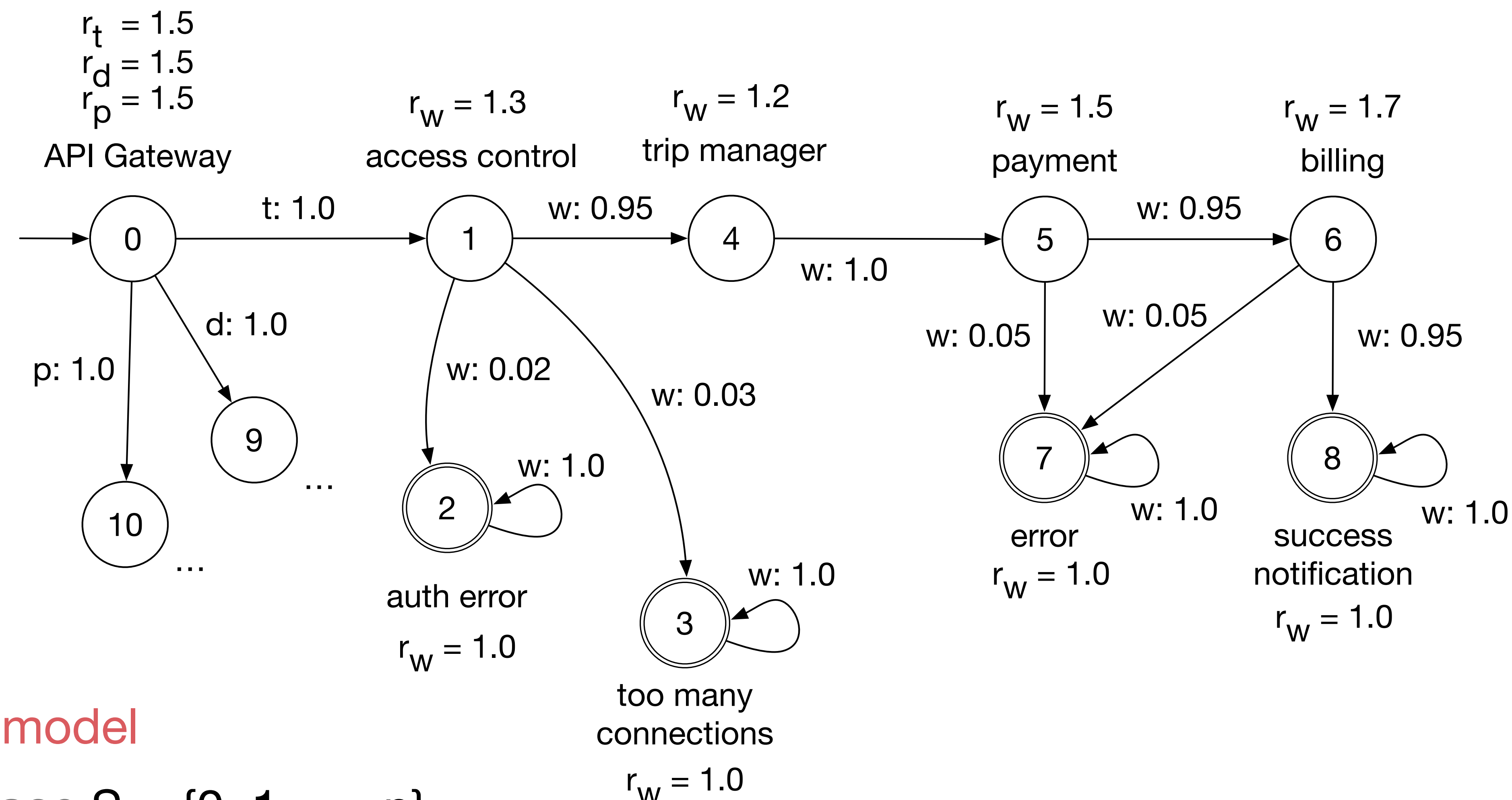
- CTMDP model
- State space S = {0, 1, …, n}

A running example: the taxi-hailing application



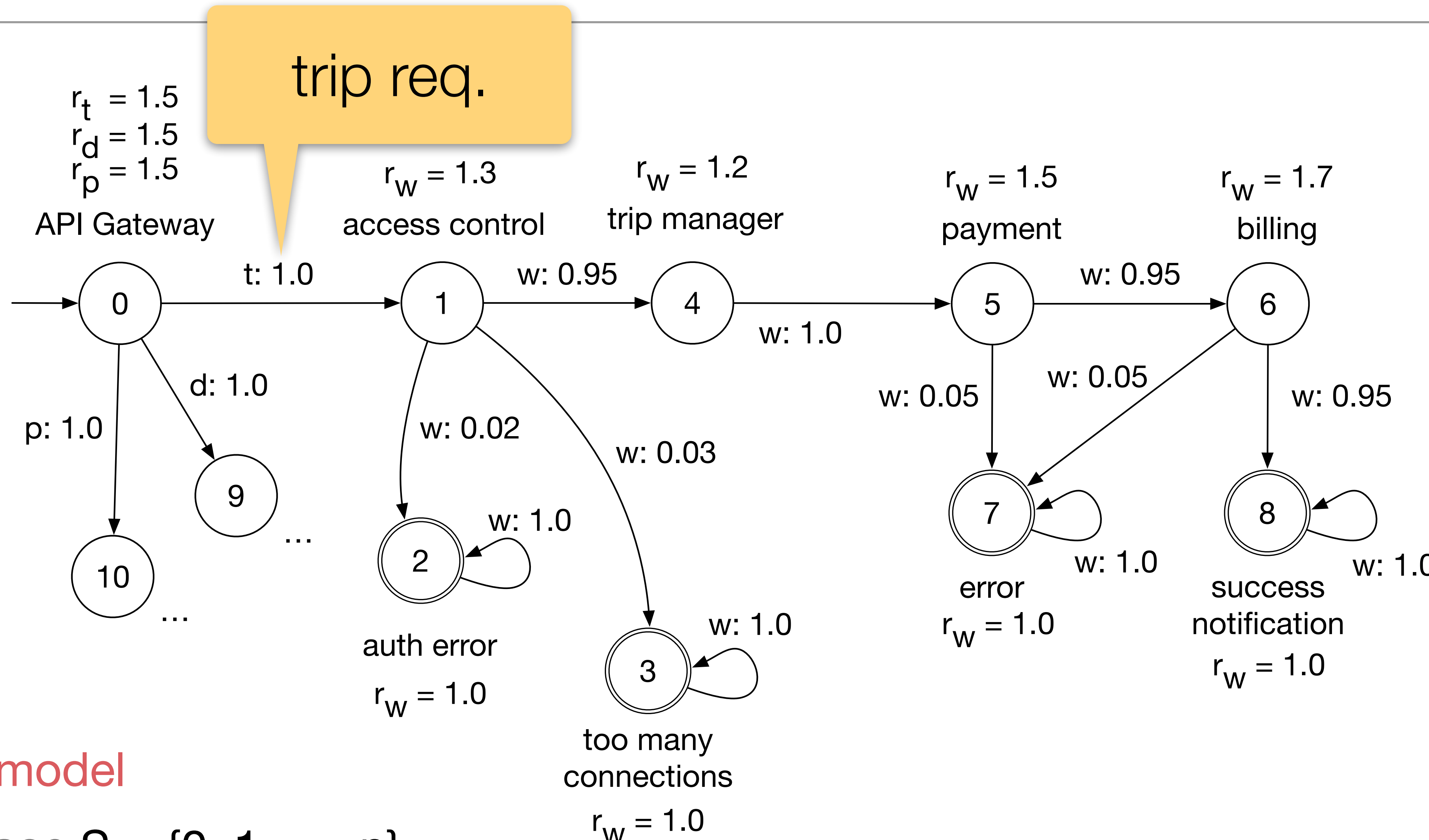
- CTMDP model
- State space S = {0, 1, …, n}
- Actions $A_i = \{a_1, \ldots, a_m\}, \forall i \in S$. *E.g., $A_0 = \{t, d, p\}$, $A_0 = \{w\}$, etc.*

A running example: the taxi-hailing application



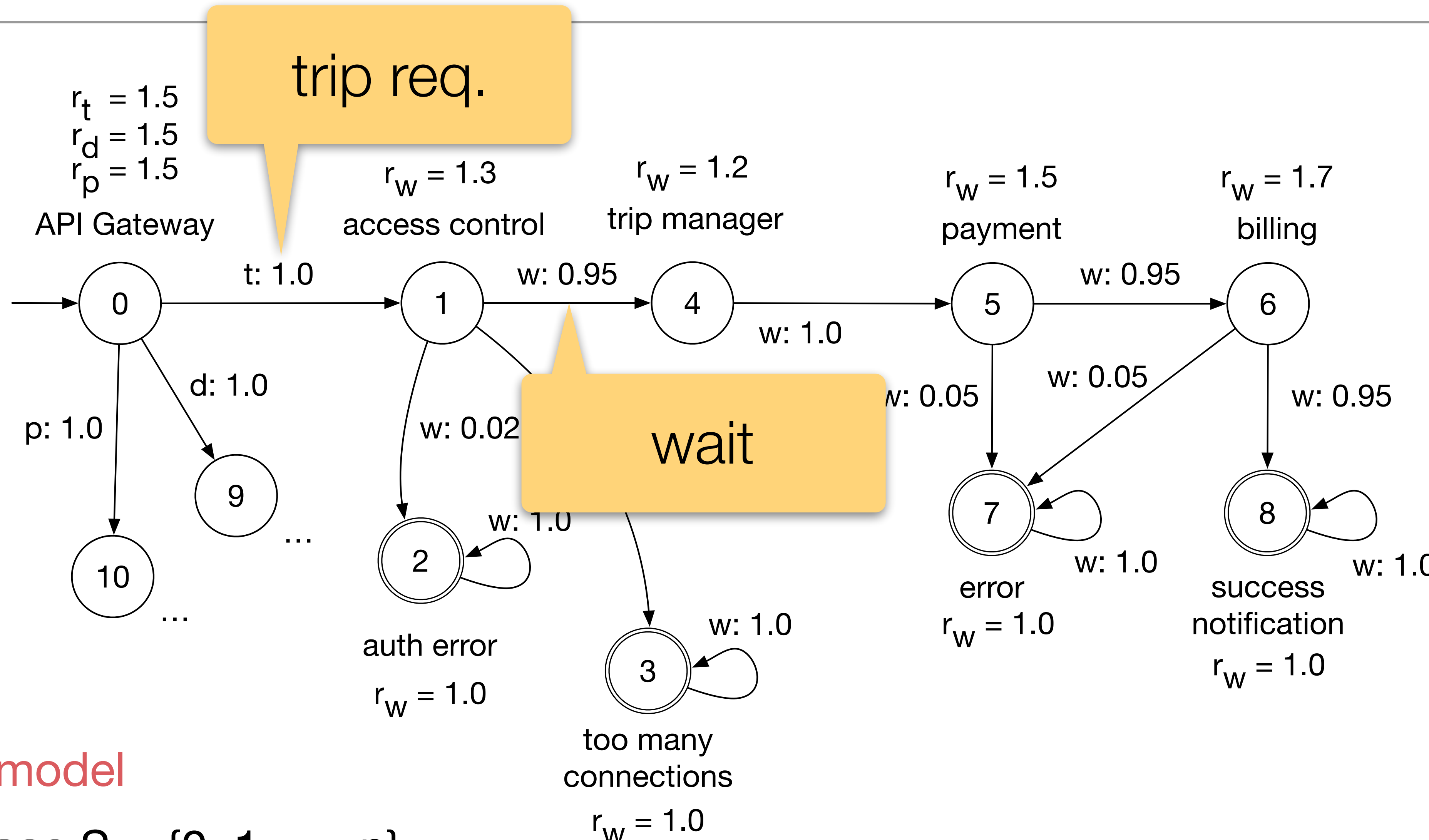
- CTMDP model
- State space S = {0, 1, …, n}
- Actions $A_i = \{a_1, \dots, a_m\}$, $\forall i \in S$. *E.g., $A_0 = \{t, d, p\}$, $A_0 = \{w\}$, etc.*
- Transition probabilities $p_a(i, j) \in [0,1]$

A running example: the taxi-hailing application



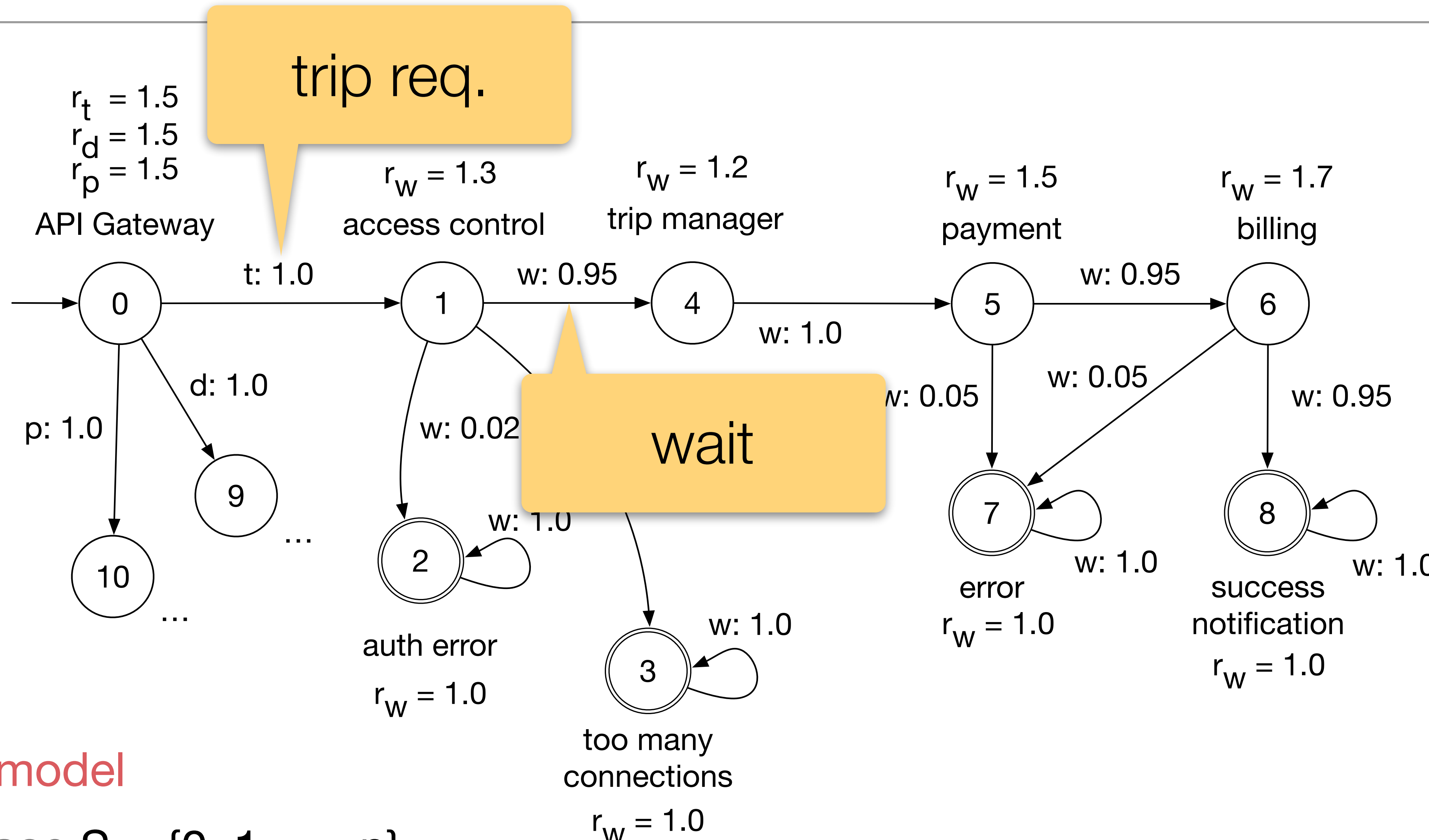
- CTMDP model
- State space S = {0, 1, …, n}
- Actions $A_i = \{a_1, \ldots, a_m\}, \forall i \in S$. *E.g., $A_0 = \{t, d, p\}$, $A_0 = \{w\}$, etc.*
- Transition probabilities $p_a(i, j) \in [0,1]$

A running example: the taxi-hailing application



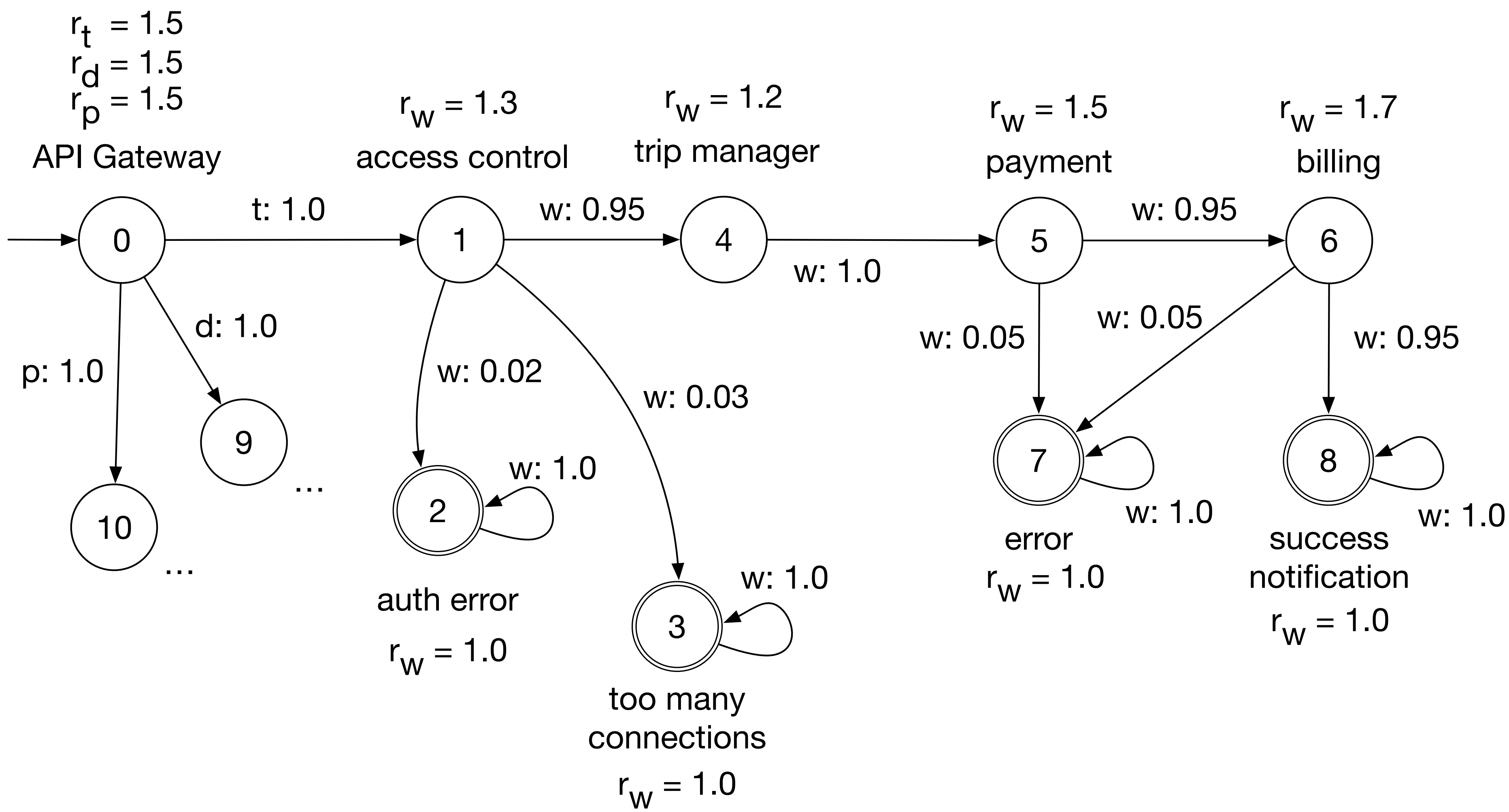
- CTMDP model
- State space S = {0, 1, …, n}
- Actions $A_i = \{a_1, \dots, a_m\}$, $\forall i \in S$. *E.g., A_0 = {t, d, p}, A_0 = {w}, etc.*
- Transition probabilities $p_a(i, j) \in [0,1]$

A running example: the taxi-hailing application

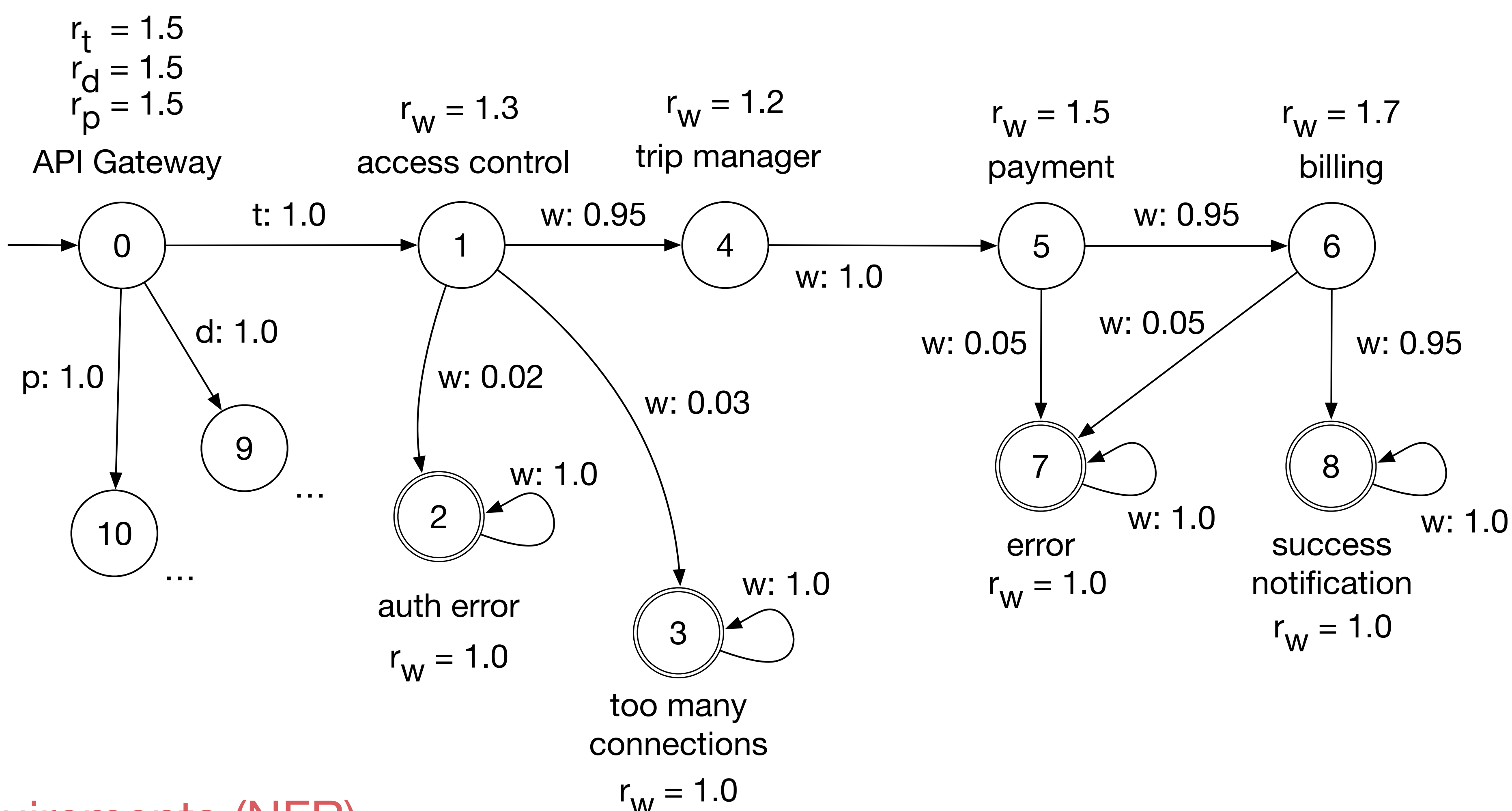


- CTMDP model
- State space S = {0, 1, …, n}
- Actions $A_i = \{a_1, \ldots, a_m\}, \forall i \in S$. *E.g., $A_0 = \{t, d, p\}$, $A_0 = \{w\}$, etc.*
- Transition probabilities $p_a(i, j) \in [0,1]$
- Exponential sojourn time $r_a(i) \in \mathbb{R}_{>0}$; Rewards $R_a(i, j) = 0, \forall i \neq j$

A running example: the taxi-hailing application

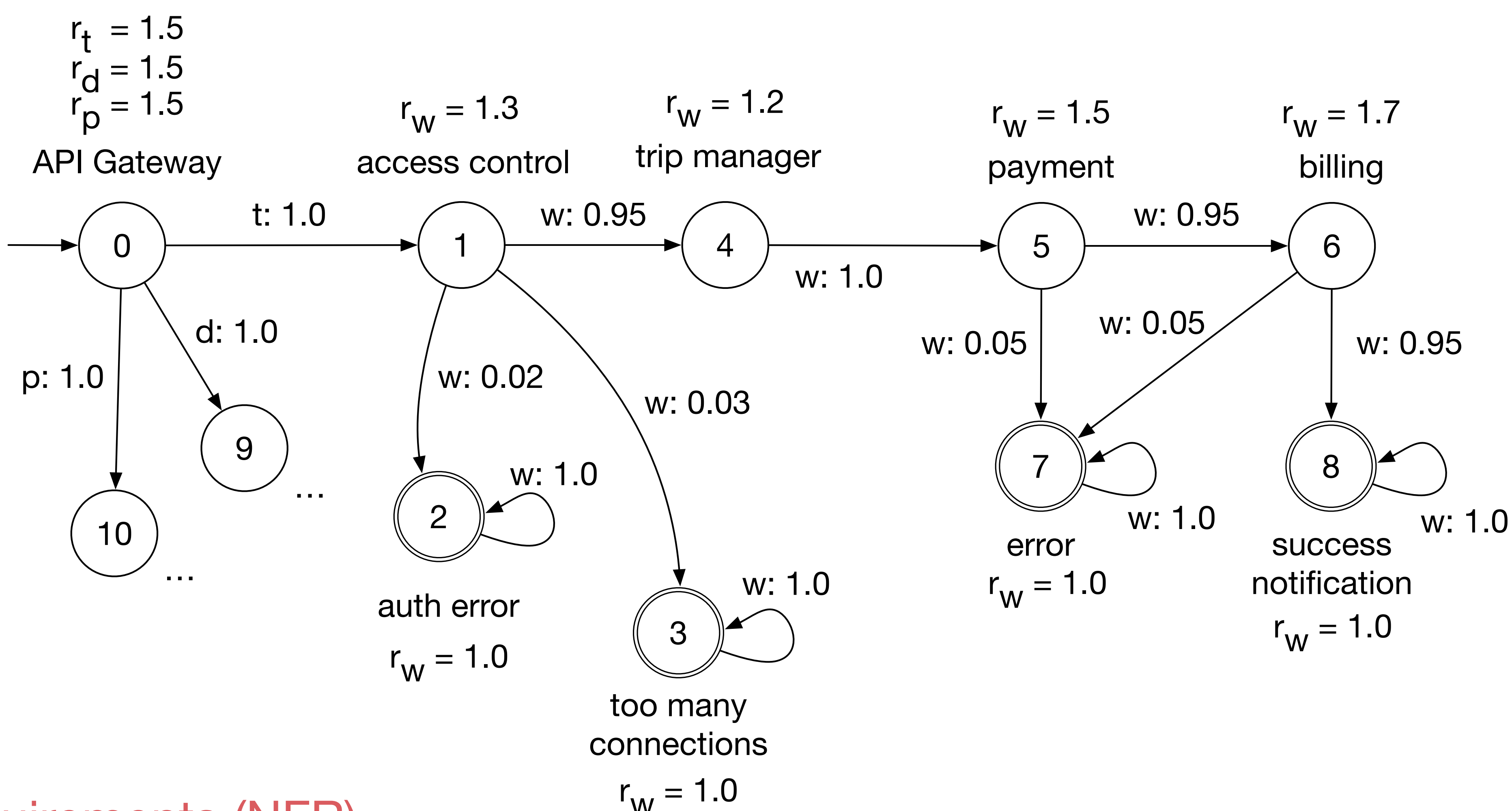


A running example: the taxi-hailing application



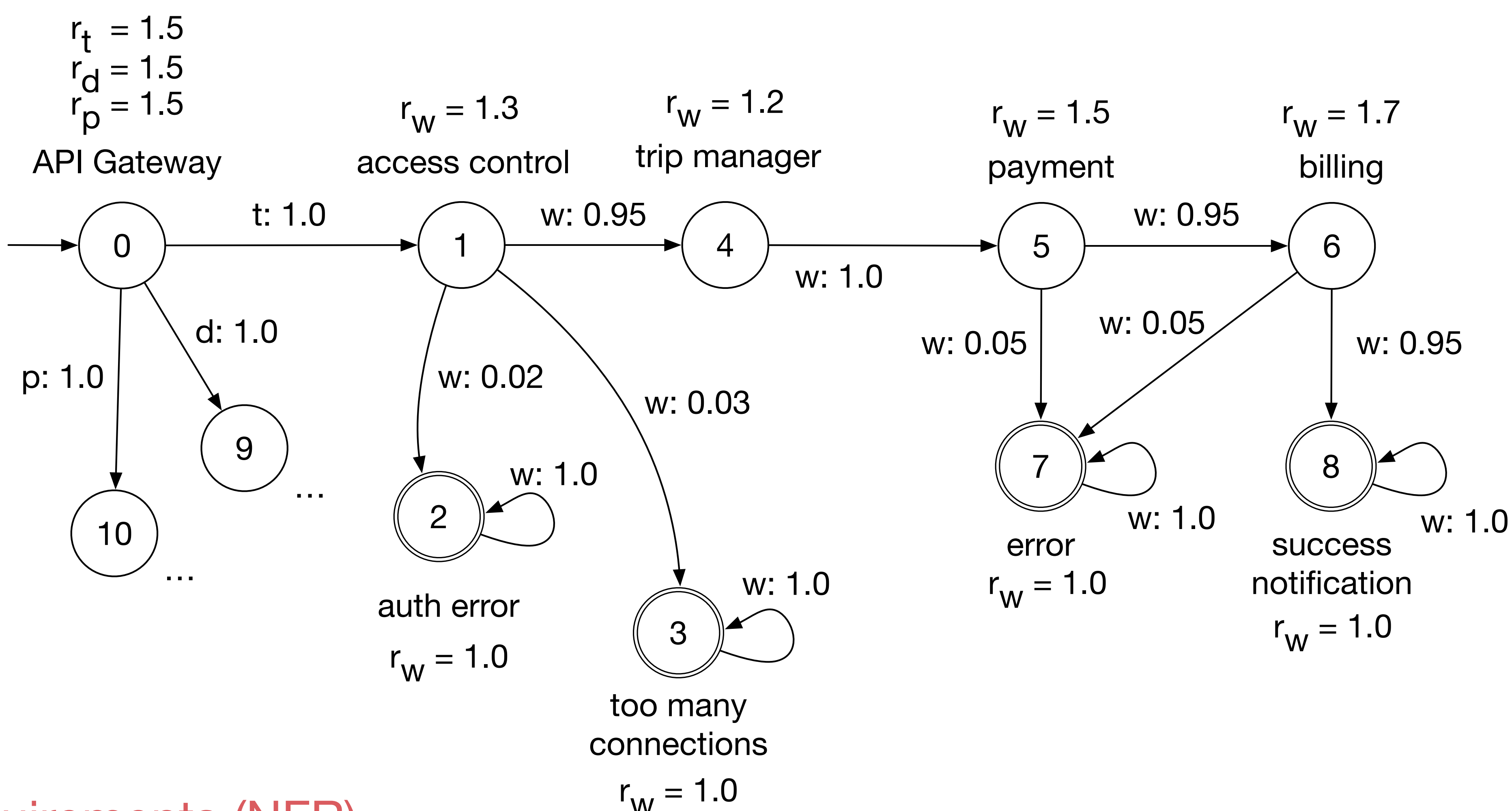
- CSL Requirements (NFR)

A running example: the taxi-hailing application



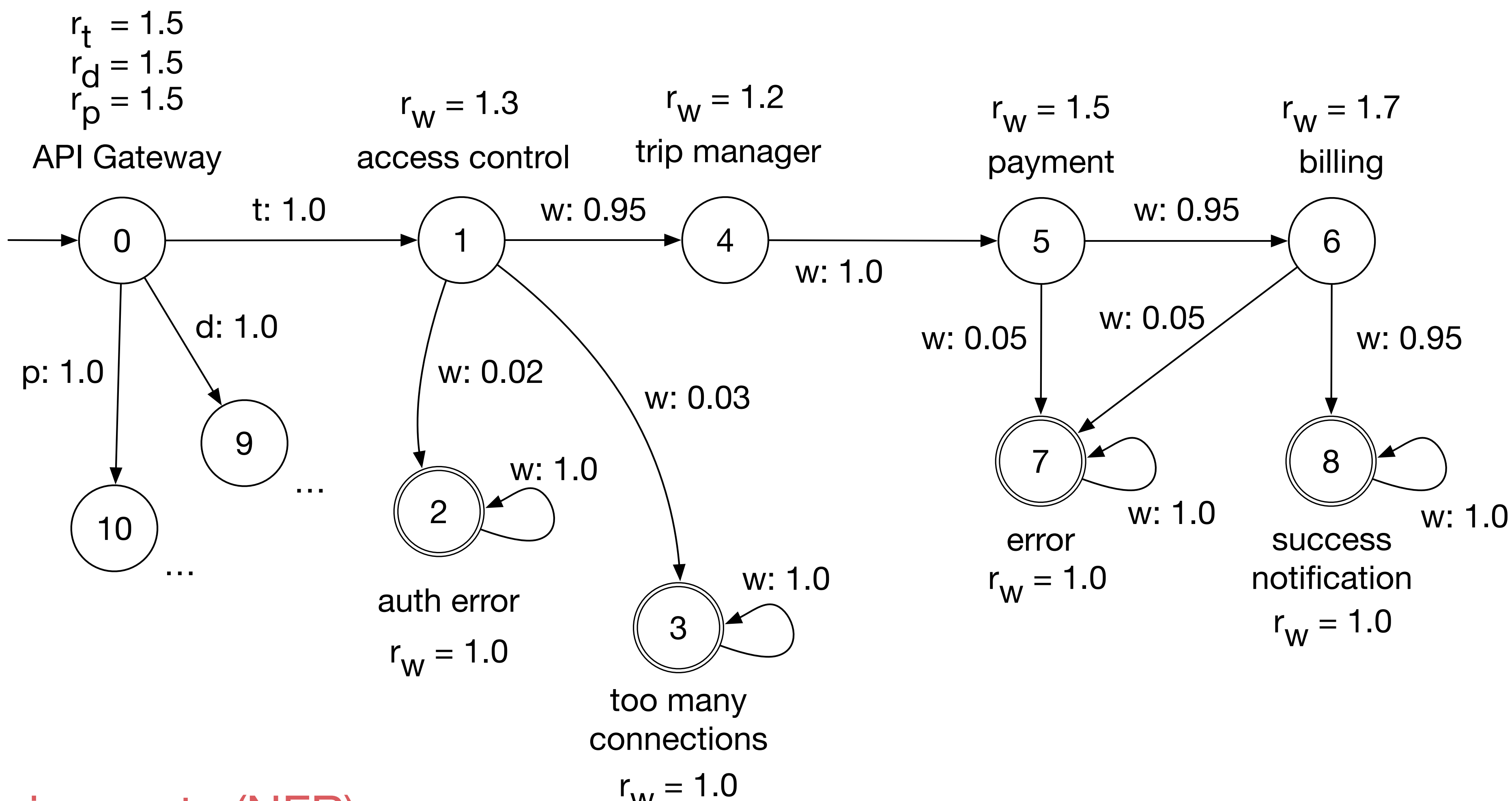
- CSL Requirements (NFR)
- (NFR1) *Reliability*: A [G "trip manager" => $P_{\geq 0.9}$ [F "notification"]]

A running example: the taxi-hailing application



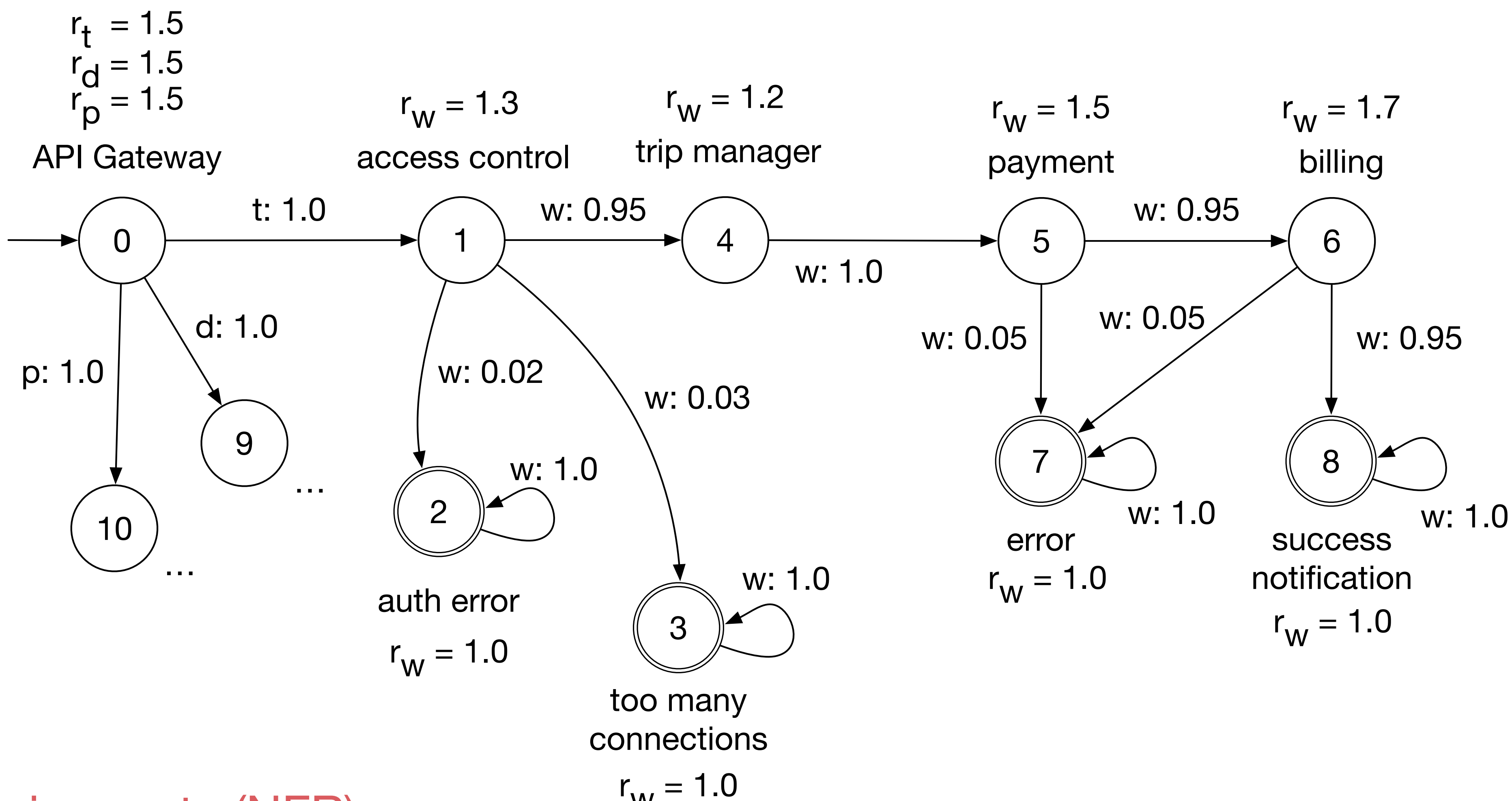
- CSL Requirements (NFR)
- (NFR1) *Reliability*: A [G “trip manager” => P$\geq_{0.9}$ [F “notification”]]
- (NFR2) *Response time*: P$>_{0.7}$ [F$_{<2.0}$ “trip manager”]

A running example: the taxi-hailing application



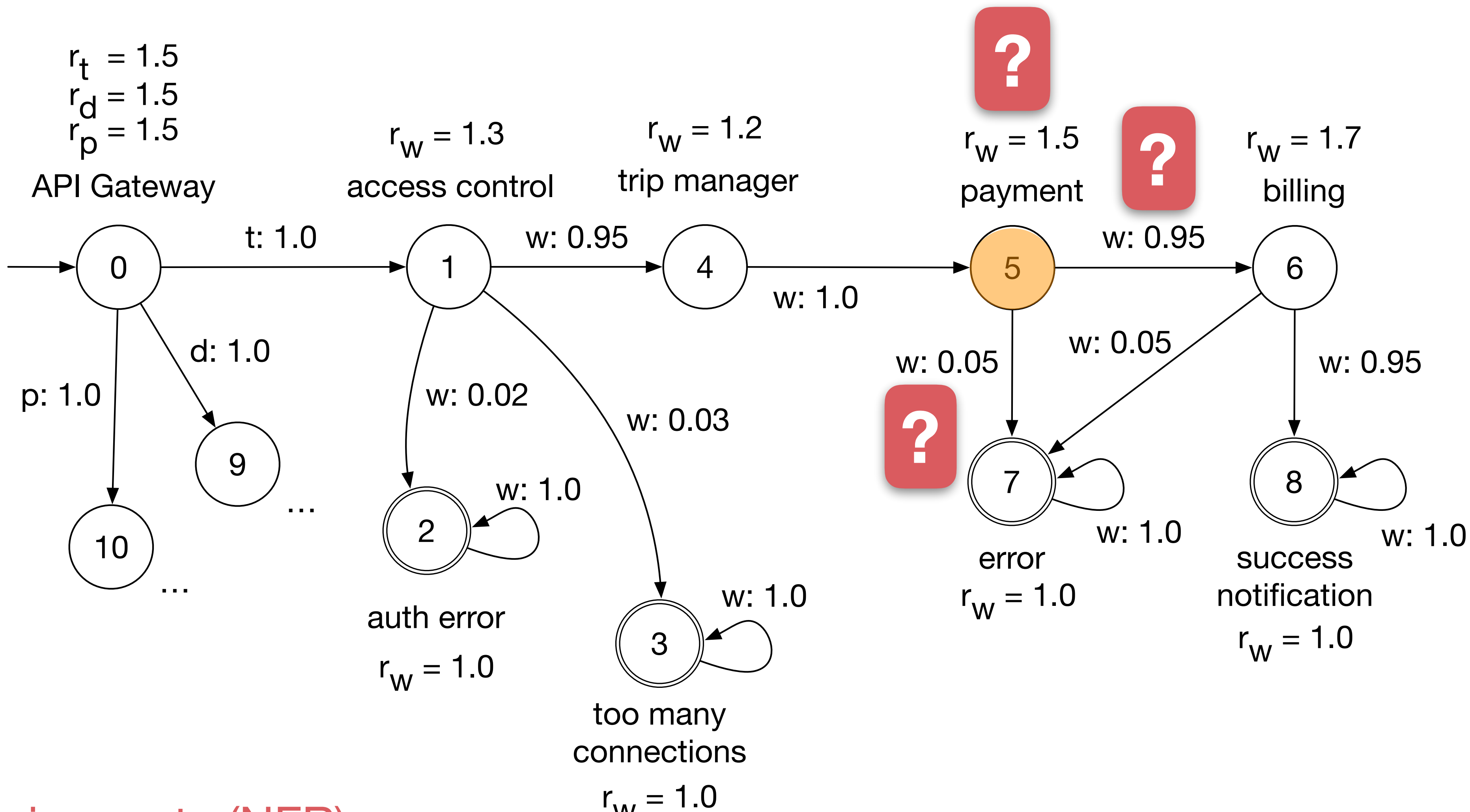
- CSL Requirements (NFR)
- (NFR1) *Reliability*: A [G “trip manager” => $P_{\geq 0.9}$ [F “notification”]]
- (NFR2) *Response time*: $P_{>0.7}$ [$F_{<2.0}$ “trip manager”]
- (NFR3) *Energy consumption*: $R_{\leq 0.5}$ [F “notification” ∨ “trip computed”]

A running example: the taxi-hailing application



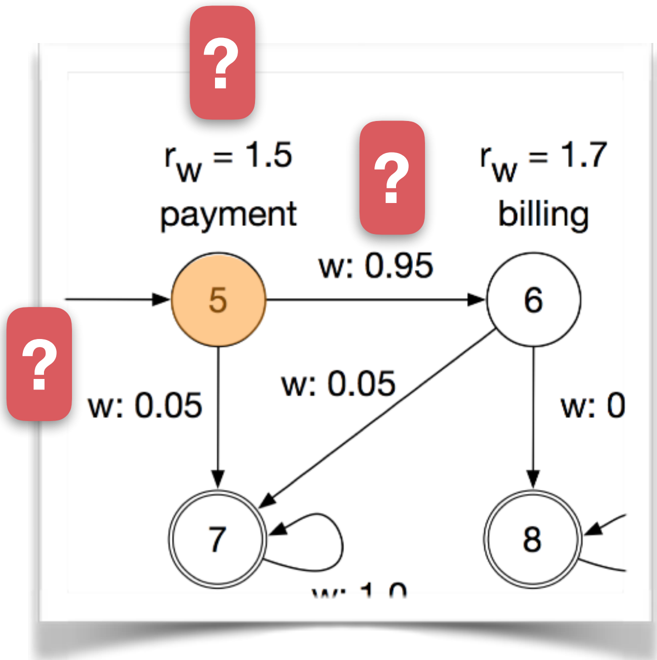
- CSL Requirements (NFR)
- (NFR1) *Reliability*: A [G “trip manager” => $P_{\geq 0.9}$ [F “notification”]] ✓
- (NFR2) *Response time*: $P_{>0.7}$ [$F_{<2.0}$ “trip manager”] ✓
- (NFR3) *Energy consumption*: $R_{\leq 0.5}$ [F “notification” ∨ “trip computed”] ✓

A running example: the taxi-hailing application



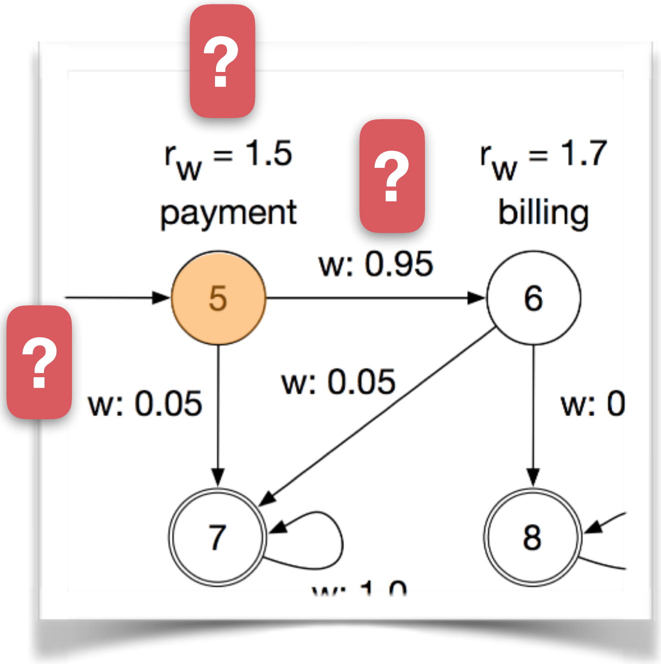
- CSL Requirements (NFR)
- (NFR1) *Reliability*: A [G “trip manager” => $P_{\geq 0.9}$ [F “notification”]] ✓ ?
- (NFR2) *Response time*: $P_{>0.7}$ [$F_{<2.0}$ “trip manager”] ✓
- (NFR3) *Energy consumption*: $R_{\leq 0.5}$ [F “notification” ∨ “trip computed”] ✓

Inference using online MBT

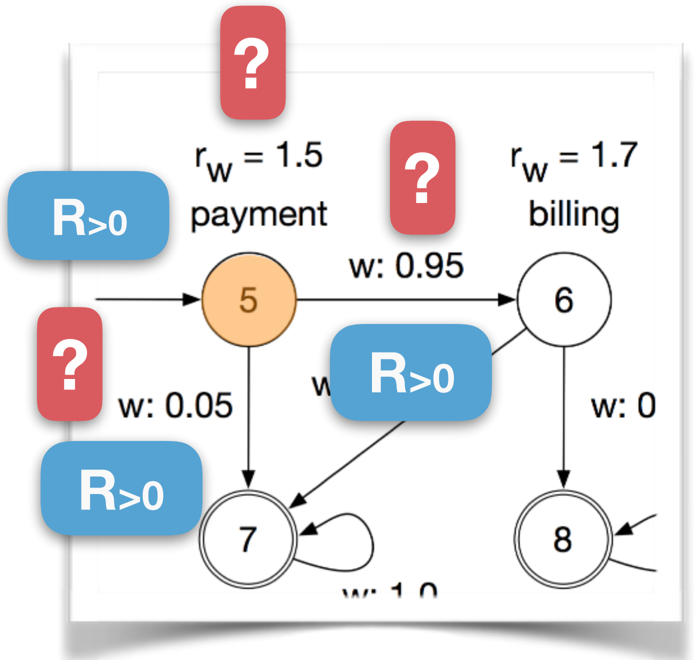


Inference using online MBT

- Priors
  - uncertain rates: $r_w(5) \sim Ga(a, b)$
  - uncertain transition probabilities: $(p_w(5,6), p_w(5,7)) \sim Dir(\alpha_1, \alpha_2)$

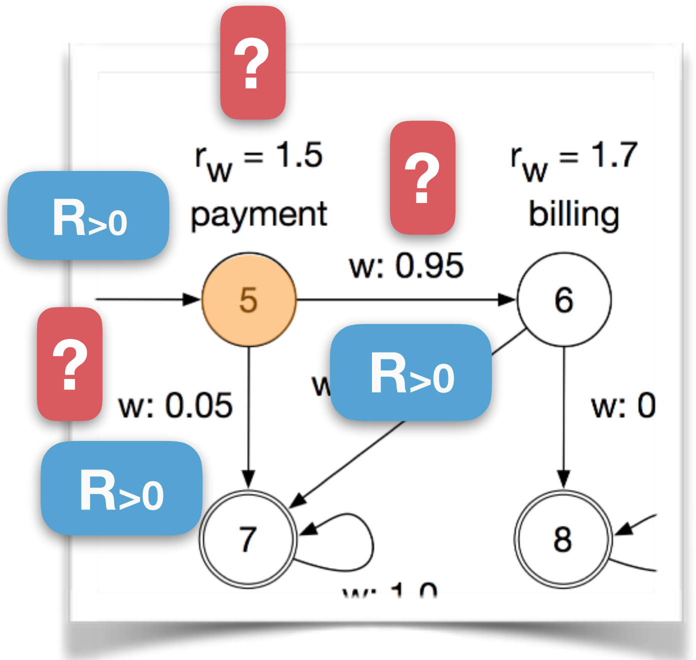


Inference using online MBT



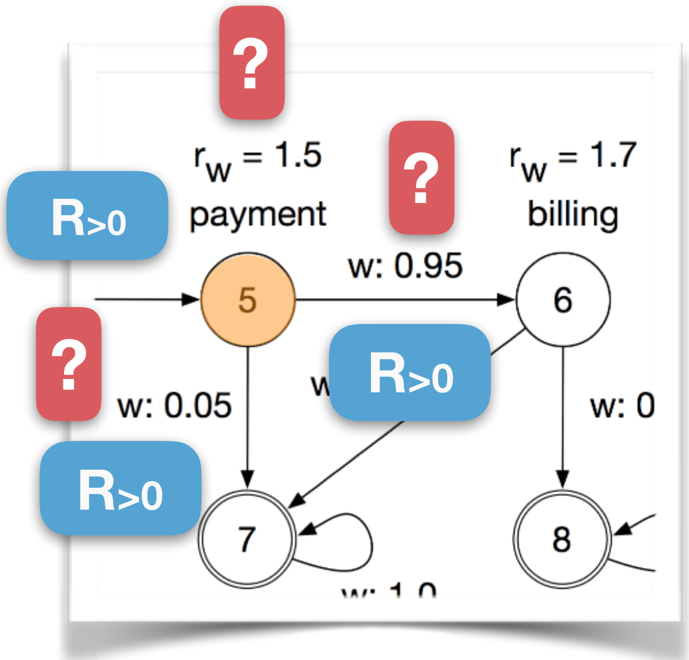
- Priors
  - uncertain rates: $r_w(5) \sim Ga(a, b)$
  - uncertain transition probabilities: $(p_w(5,6), p_w(5,7)) \sim Dir(\alpha_1, \alpha_2)$
- Policy
  - we set a reward $R_{>0}$ attached to uncertain components
  - we compute a **reward dependent policy** π that maximizes some cumulative function on the rewards: $\sum_{i=0}^{\infty} \gamma^t R(s_i, s_{i+1})$

Inference using online MBT



- Priors
  - uncertain rates: $r_w(5) \sim Ga(a, b)$
  - uncertain transition probabilities: $(p_w(5,6), p_w(5,7)) \sim Dir(\alpha_1, \alpha_2)$
- Policy
  - we set a reward R>0 attached to uncertain components
  - we compute a **reward dependent policy** π that maximizes some cumulative function on the rewards: $\sum_{i=0}^{\infty} \gamma^t R(s_i, s_{i+1})$
- Online MBT
  - it guides the execution of the system by applying the policy π

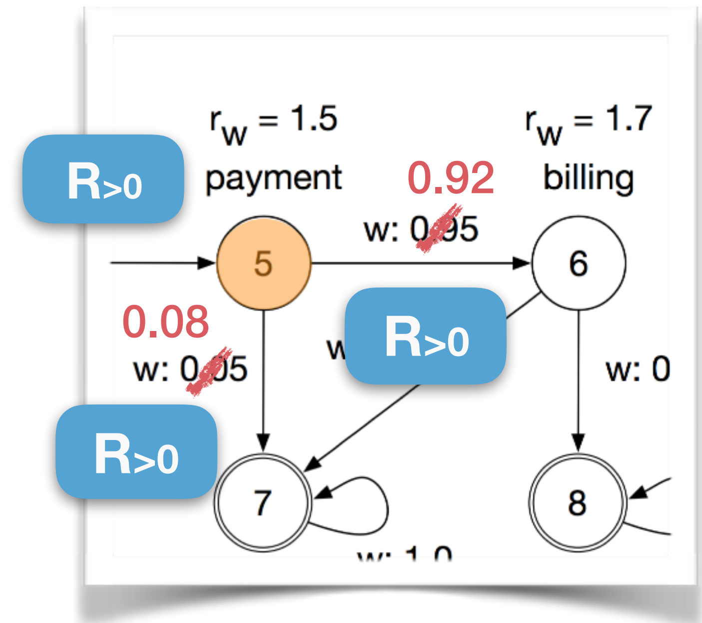
Inference using online MBT



- Priors
  - uncertain rates: $r_W(5) \sim Ga(a, b)$
  - uncertain transition probabilities: $(p_W(5,6), p_W(5,7)) \sim Dir(\alpha_1, \alpha_2)$
- Policy
  - we set a reward $R_{>0}$ attached to uncertain components
  - we compute a **reward dependent policy** π that maximizes some cumulative function on the rewards: $\sum_{i=0}^{\infty} \gamma^t R(s_i, s_{i+1})$
- Online MBT
  - it guides the execution of the system by applying the policy π
- Inference
  - updating rules: $r_W(5)|y^e \sim Ga(a+n_5, b+T_5)$, $(p_W(5,6), p_W(5,7))|y^e \sim Dir(\alpha_1+n_{5,6}, \alpha_2+n_{5,7})$
  - summarization through the **posterior modes**

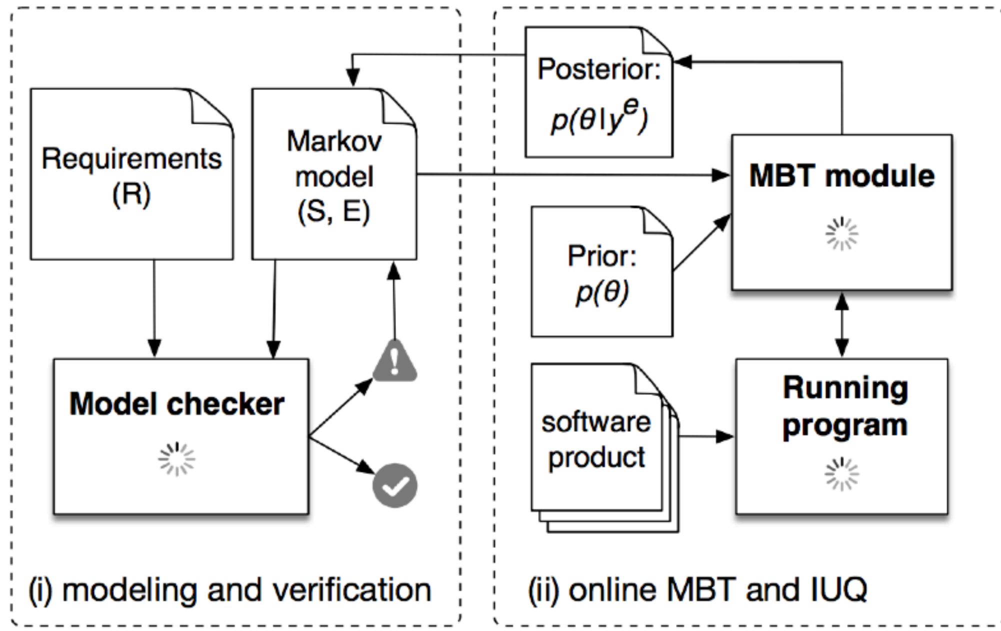
Inference using online MBT

- Priors
  - uncertain rates: $r_w(5) \sim Ga(a, b)$
  - uncertain transition probabilities: $(p_w(5,6), p_w(5,7)) \sim Dir(\alpha_1, \alpha_2)$
- Policy
  - we set a reward $R_{>0}$ attached to uncertain components
  - we compute a **reward dependent policy** π that maximizes some cumulative function on the rewards: $\sum_{i=0}^{\infty} \gamma^t R(s_i, s_{i+1})$
- Online MBT
  - it guides the execution of the system by applying the policy π
- Inference
  - updating rules: $r_w(5)|y^e \sim Ga(a+n_5, b+T_5)$, $(p_w(5,6), p_w(5,7))|y^e \sim Dir(\alpha_1+n_{5,6}, \alpha_2+n_{5,7})$
  - summarization through the **posterior modes**

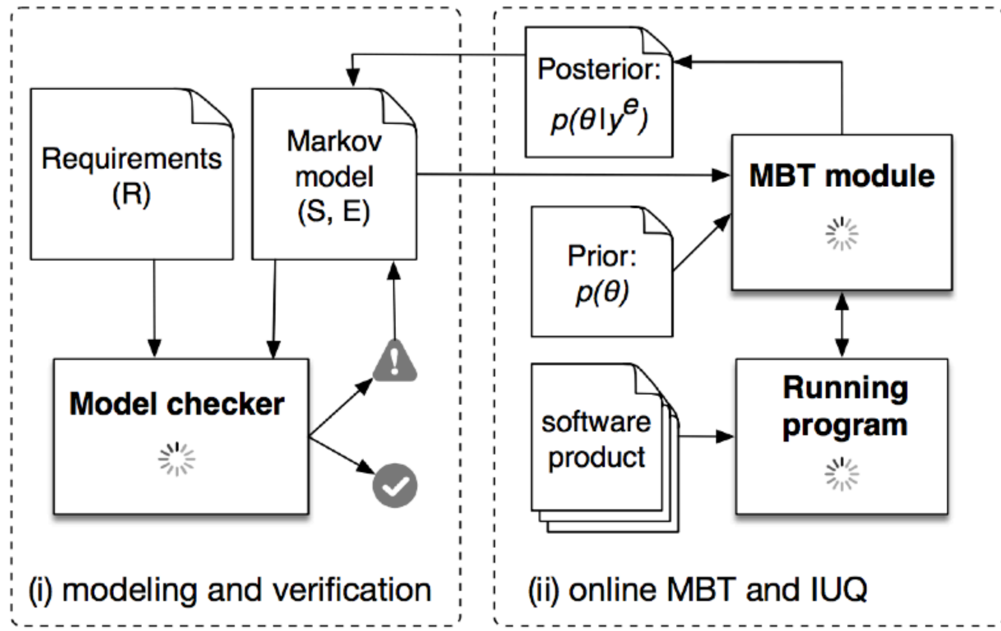


(NFR1) ✗

Toolchain

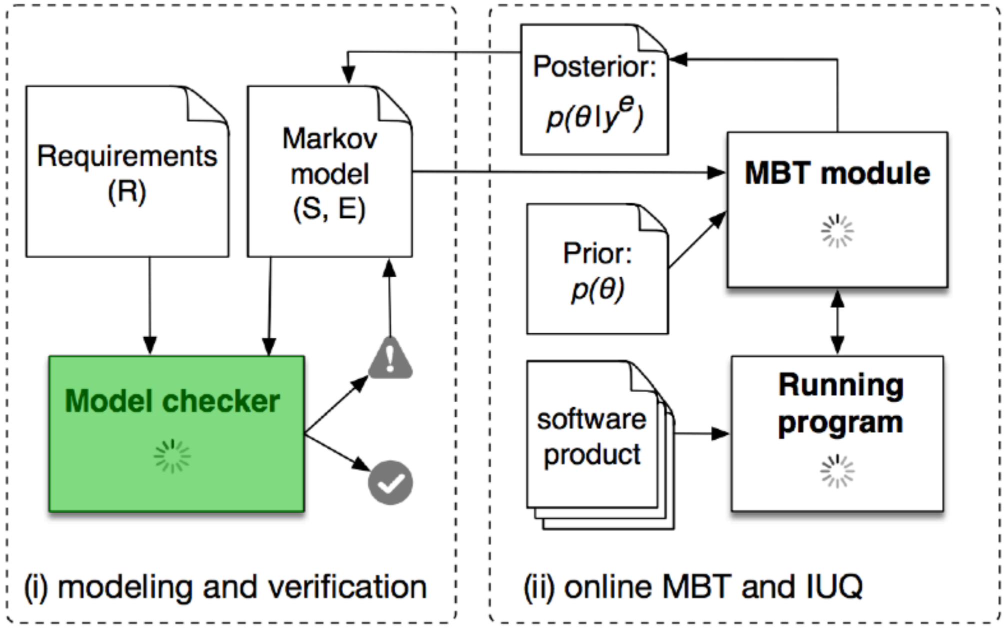

Requirements (R)
Markov model (S, E)
Model checker
Posterior: $p(\theta | y^e)$
MBT module
Prior: $p(\theta)$
software product
Running program
(i) modeling and verification
(ii) online MBT and IUQ

Toolchain



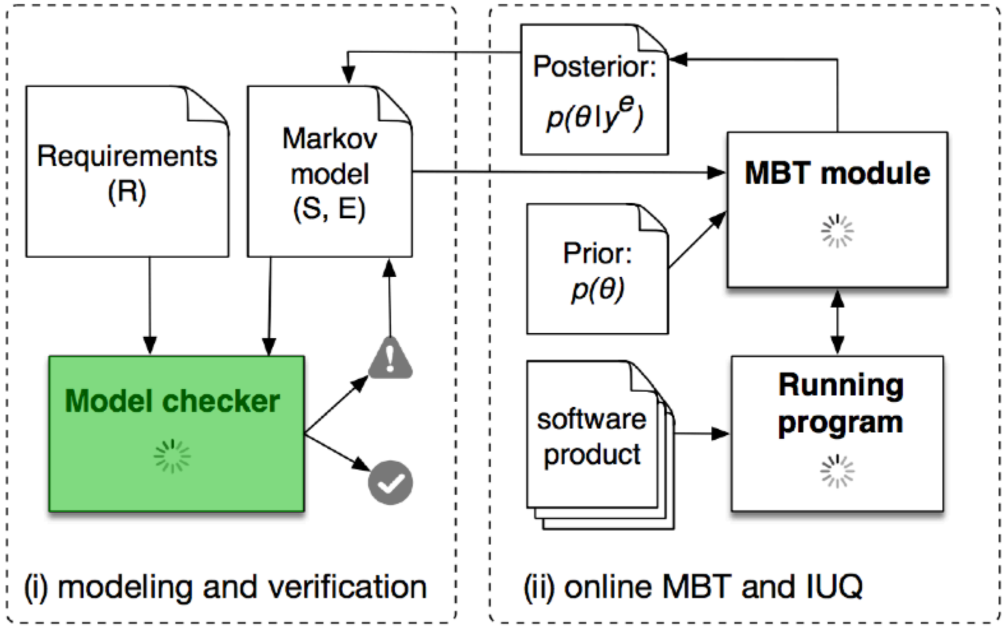
- (i) modeling and verification

Toolchain



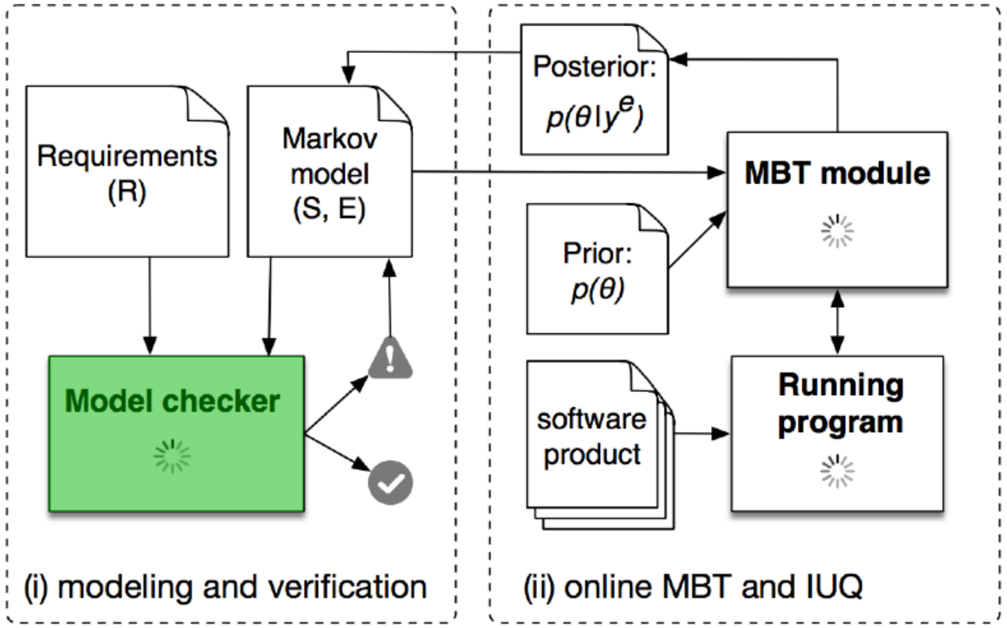
- (i) modeling and verification
  - probabilistic model checking using the *MRMC* model checker

Toolchain



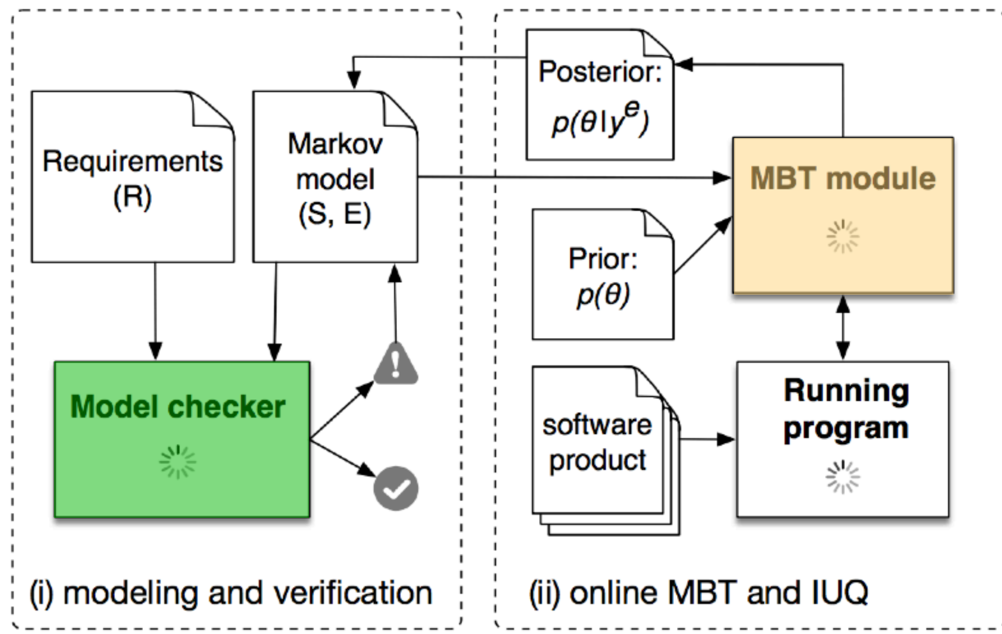
- (i) modeling and verification
  - probabilistic model checking using the *MRMC* model checker
  - a simple DSL aids the definition of the CTMDP model and automatically map actions to *observable/controllable* methods of the implementation (i.e., a Java program)

Toolchain



- (i) modeling and verification
  - probabilistic model checking using the *MRMC* model checker
  - a simple DSL aids the definition of the CTMDP model and automatically map actions to *observable/controllable* methods of the implementation (i.e., a Java program)
- (ii) online MBT and IUQ

Toolchain



- (i) modeling and verification
  - probabilistic model checking using the *MRMC* model checker
  - a simple DSL aids the definition of the CTMDP model and automatically map actions to *observable/controllable* methods of the implementation (i.e., a Java program)
- (ii) online MBT and IUQ
  - we are in the process of validating the online MBT module with a realistic case-study deployed on a cloud infrastructure

Conclusion

Conclusion

- Key points

Conclusion

- Key points
  - we explicitly take into account the **uncertainty** in software development

Conclusion

- Key points
  - we explicitly take into account the uncertainty in software development
  - we mainly focus on the IUQ during the development life cycle by combining online MBT and Bayesian reasoning

Conclusion

- Key points
  - we explicitly take into account the uncertainty in software development
  - we mainly focus on the IUQ during the development life cycle by combining online MBT and Bayesian reasoning
- Pros

Conclusion

- Key points
  - we explicitly take into account the **uncertainty** in software development
  - we mainly focus on the **IUQ** during the development life cycle by combining **online MBT** and **Bayesian reasoning**
- Pros
  - the underlying probabilistic representation and the inference approach makes our technique **robust** to spurious/unreliable observations

Conclusion

- Key points
  - we explicitly take into account the **uncertainty** in software development
  - we mainly focus on the **IUQ** during the development life cycle by combining **online MBT** and **Bayesian reasoning**
- Pros
  - the underlying probabilistic representation and the inference approach makes our technique **robust** to spurious/unreliable observations
  - the technique is **efficient** and suitable for runtime verification

Conclusion

- Key points
  - we explicitly take into account the **uncertainty** in software development
  - we mainly focus on the **IUQ** during the development life cycle by combining **online MBT** and **Bayesian reasoning**
- Pros
  - the underlying probabilistic representation and the inference approach makes our technique **robust** to spurious/unreliable observations
  - the technique is **efficient** and suitable for runtime verification
- Next

Conclusion

- Key points
  - we explicitly take into account the **uncertainty** in software development
  - we mainly focus on the **IUQ** during the development life cycle by combining **online MBT** and **Bayesian reasoning**
- Pros
  - the underlying probabilistic representation and the inference approach makes our technique **robust** to spurious/unreliable observations
  - the technique is **efficient** and suitable for runtime verification
- Next
  - systematic evaluation of the approach on realistic case studies

Conclusion

- Key points
  - we explicitly take into account the **uncertainty** in software development
  - we mainly focus on the **IUQ** during the development life cycle by combining **online MBT** and **Bayesian reasoning**
- Pros
  - the underlying probabilistic representation and the inference approach makes our technique **robust** to spurious/unreliable observations
  - the technique is **efficient** and suitable for runtime verification
- Next
  - systematic evaluation of the approach on realistic case studies
  - allow to mix the uncertainty-based sampling method with user-defined strategies based on decrementing/fixed weights associated with controllable actions

Thank you

Questions?

Matteo Camilli, Angelo Gargantini, Patrizia Scandurra, Carlo Bellettini

matteo.camilli@unimi.it
http://camilli.di.unimi.it