Queueing Network Modeling Patterns for Reliable and Unreliable Publish/Subscribe Protocols

Georgios Bouloukakis[1,3], Ajay Kattepur[2], Nikolaos Georgantas[3] & Valérie Issarny[3]

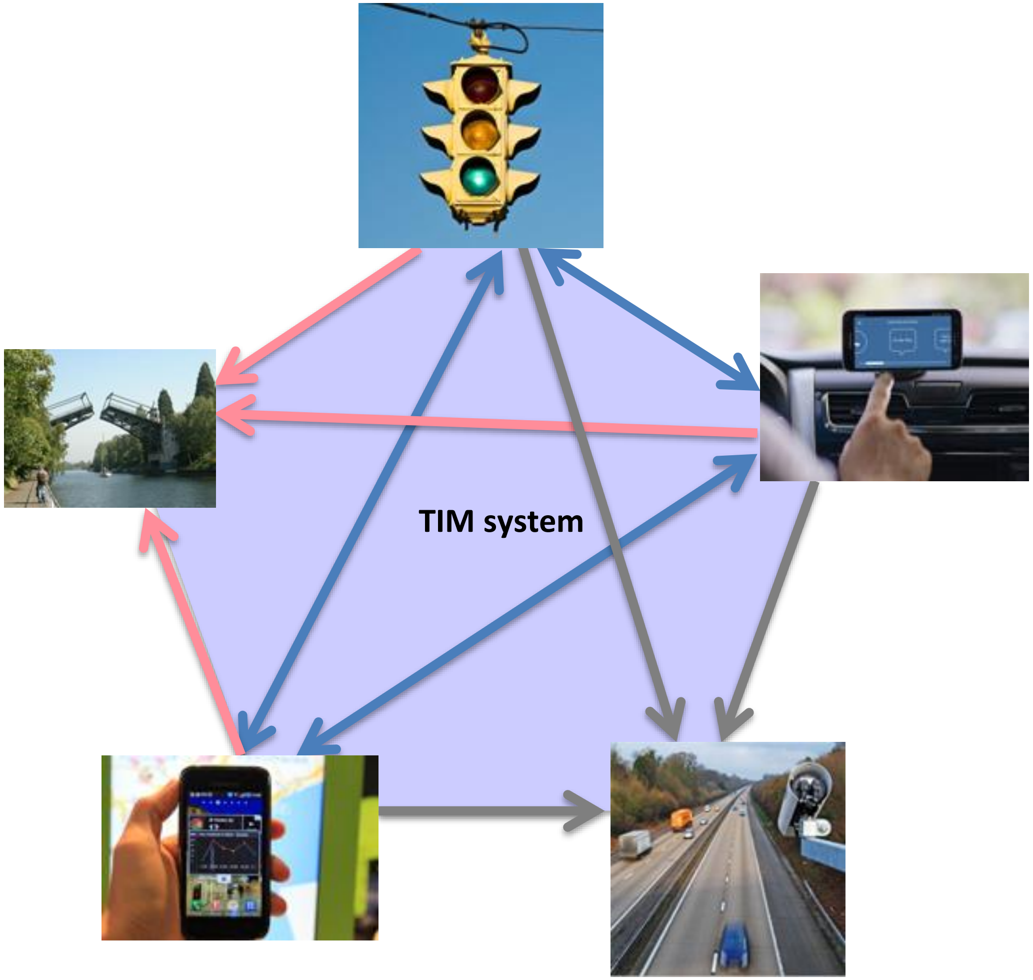
New York, USA, November 2018

15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)

[1]Donald Bren School of Information and Computer Sciences, UC Irvine, USA
[2]Embedded Systems & Robotics, TCS Research & Innovation, India
[3]MiMove team, Inria Paris, France

Motivation

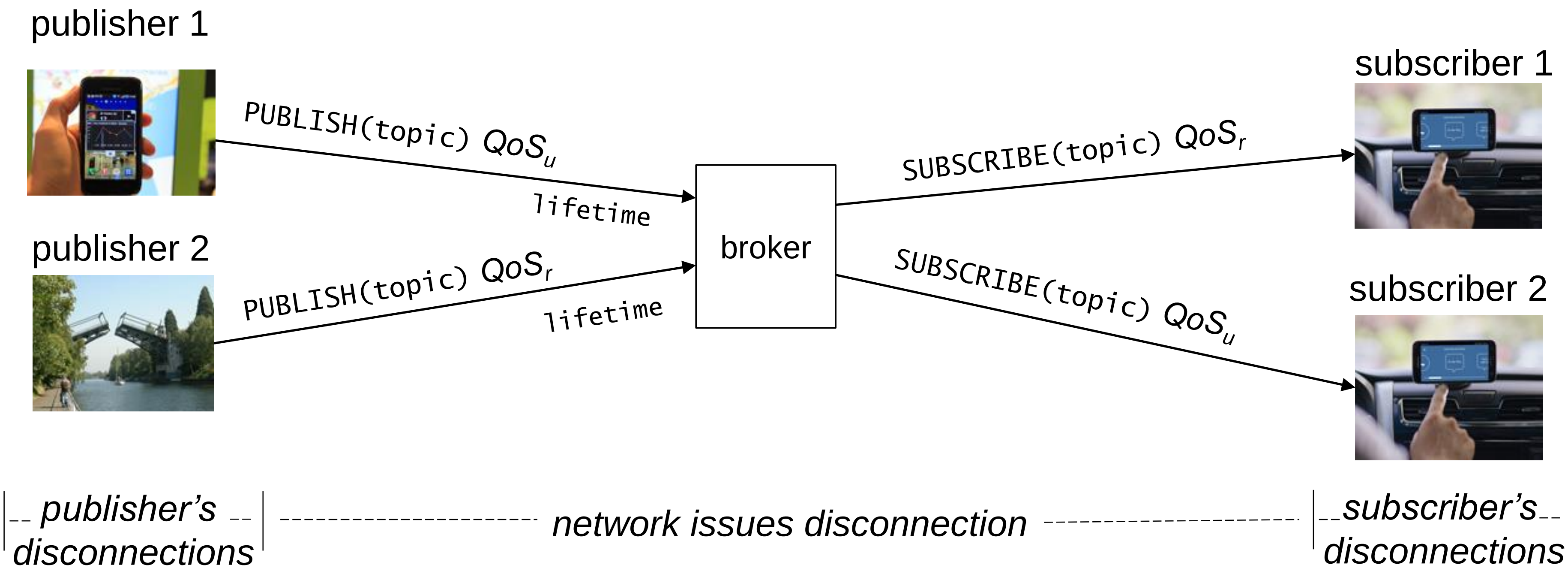
- Traffic Information Management (TIM) system:



Heterogeneous

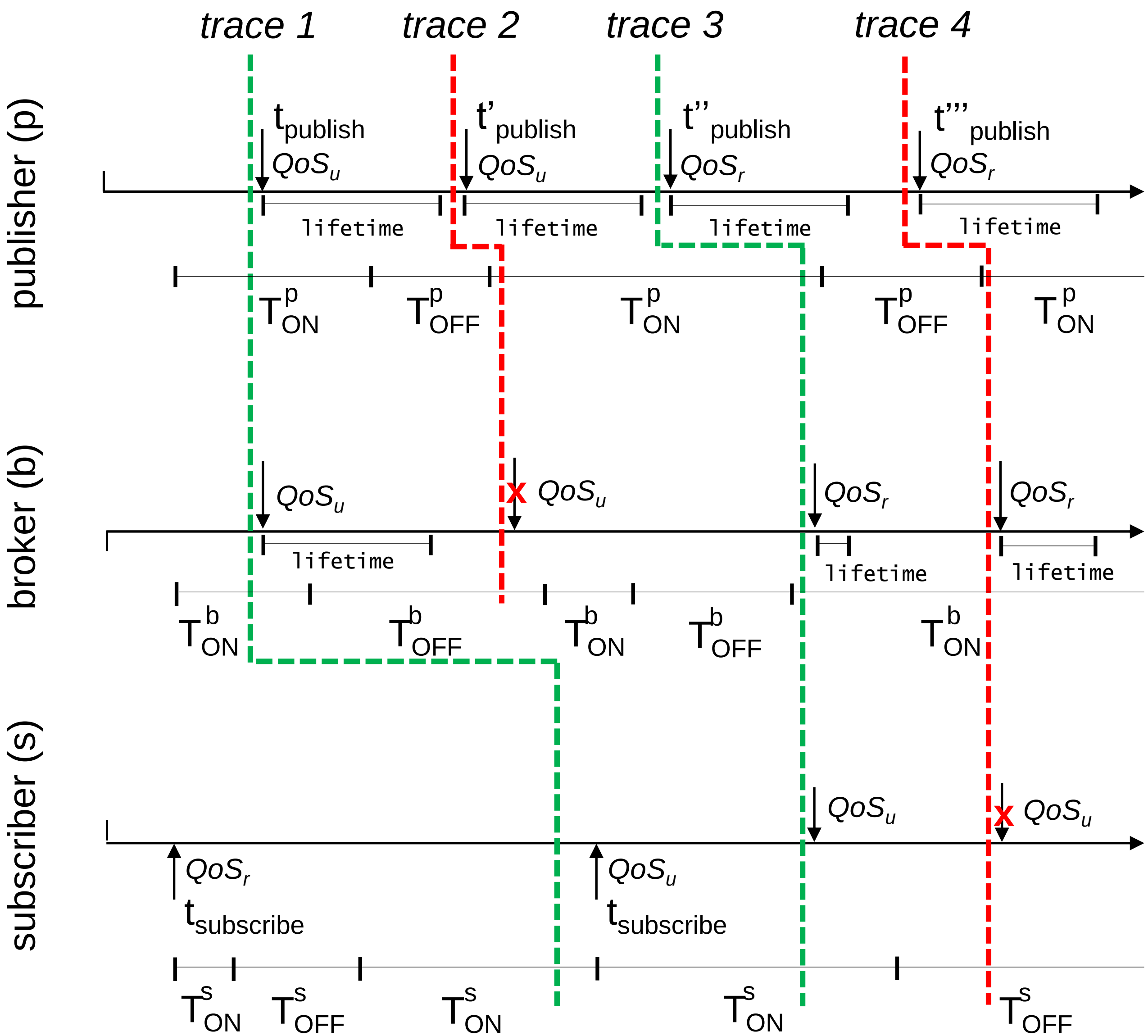
Dynamic

Publish/Subscribe (pub/sub) QoS features



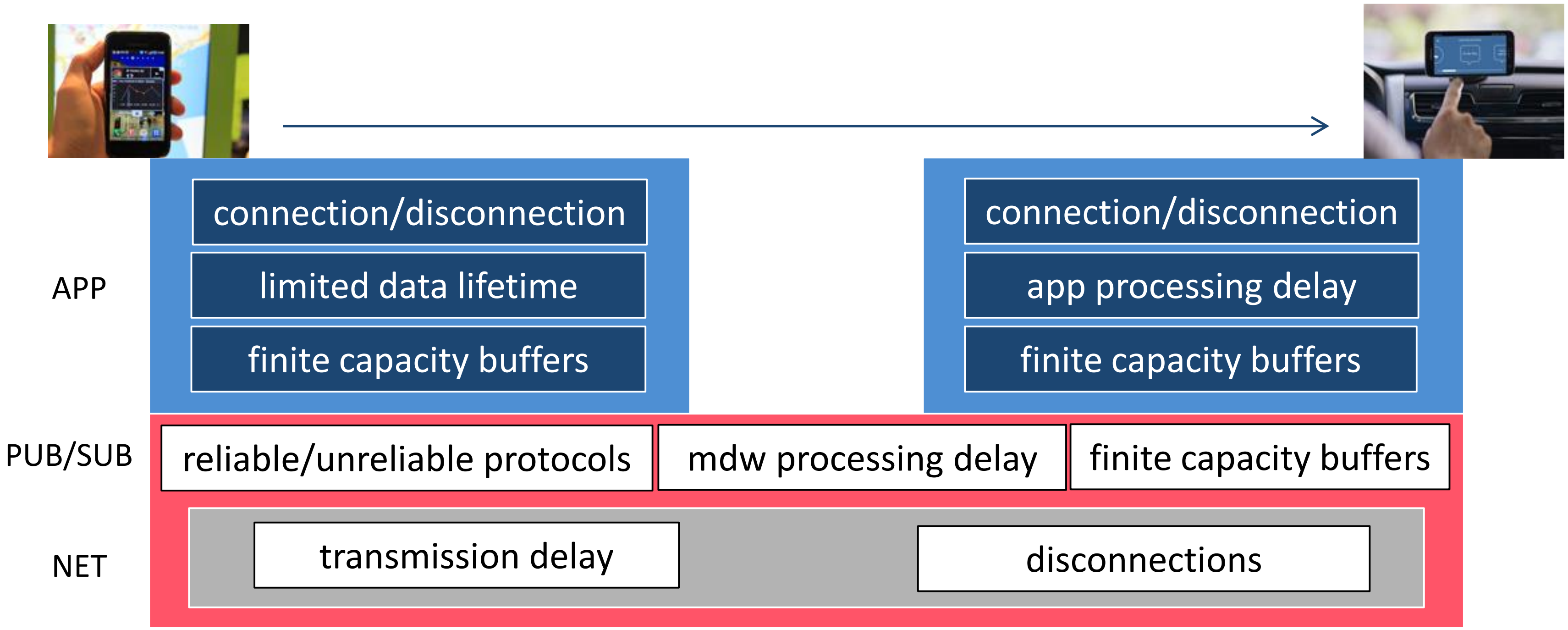
- Protocols and APIs: MQTT, AMQP, JMS, …
- IoT applications often employ such protocols and their QoS features:
  1. How to evaluate the peers' end-to-end performance?
  2. How to enable system parameter tuning to IoT applications running over pub/sub?

Analysis of Pub/Sub Interactions



Pub/Sub Interactions across Multiple Layers

- We introduce a performance modeling pattern (PerfMP) with realistic constraints found across multiple layers in the IoT

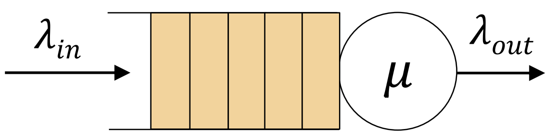


- System designers can derive values for end-to-end response times and delivery success rates

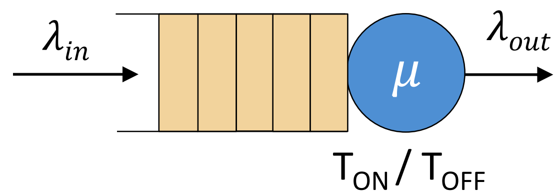
Base queueing models for Pub/Sub interactions

- We model the end-to-end path of a pub/sub interaction by using a combination of different types of queueing models

continuous queue

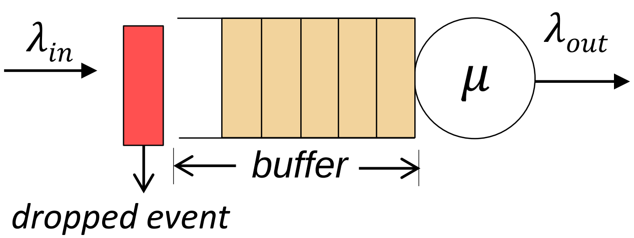


intermittent (ON/OFF) queue[1,2]

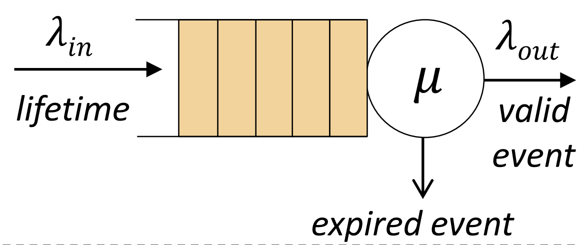


- Additional features:

finite capacity queue



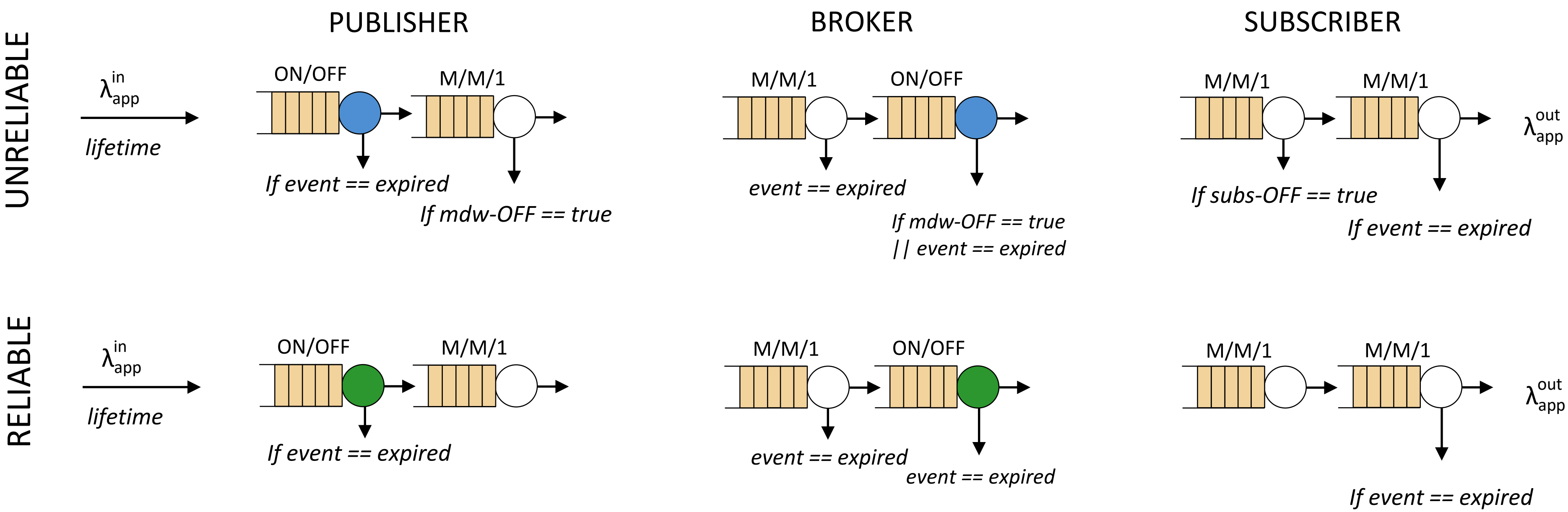
messages exp. queue



[1] G. Bouloukakis et al., ICC, 2017
[2] G. Bouloukakis et al., ICPE, 2017

PerfMP for Pub/Sub Interactions

- We model **reliable** or **unreliable** interactions by using our queueing models



Publisher's or subscriber's disconnections

End-to-end disconnection pattern between publisher-broker or broker-subscriber

Evaluation Results

- JINQS (Java Implementation of a Network-of-Queues Simulation):
  - open source simulator for building queueing networks
- We extend JINQS to implement:
  - ON/OFF queue, reliable/unreliable event transmission, other features
  - Our proposed PerfMP
- Evaluate the trade-off between response times delivery success rates for numerous reliable/unreliable interactions
- System parameter tuning for TIM

p reliable to *s* reliable: success rates

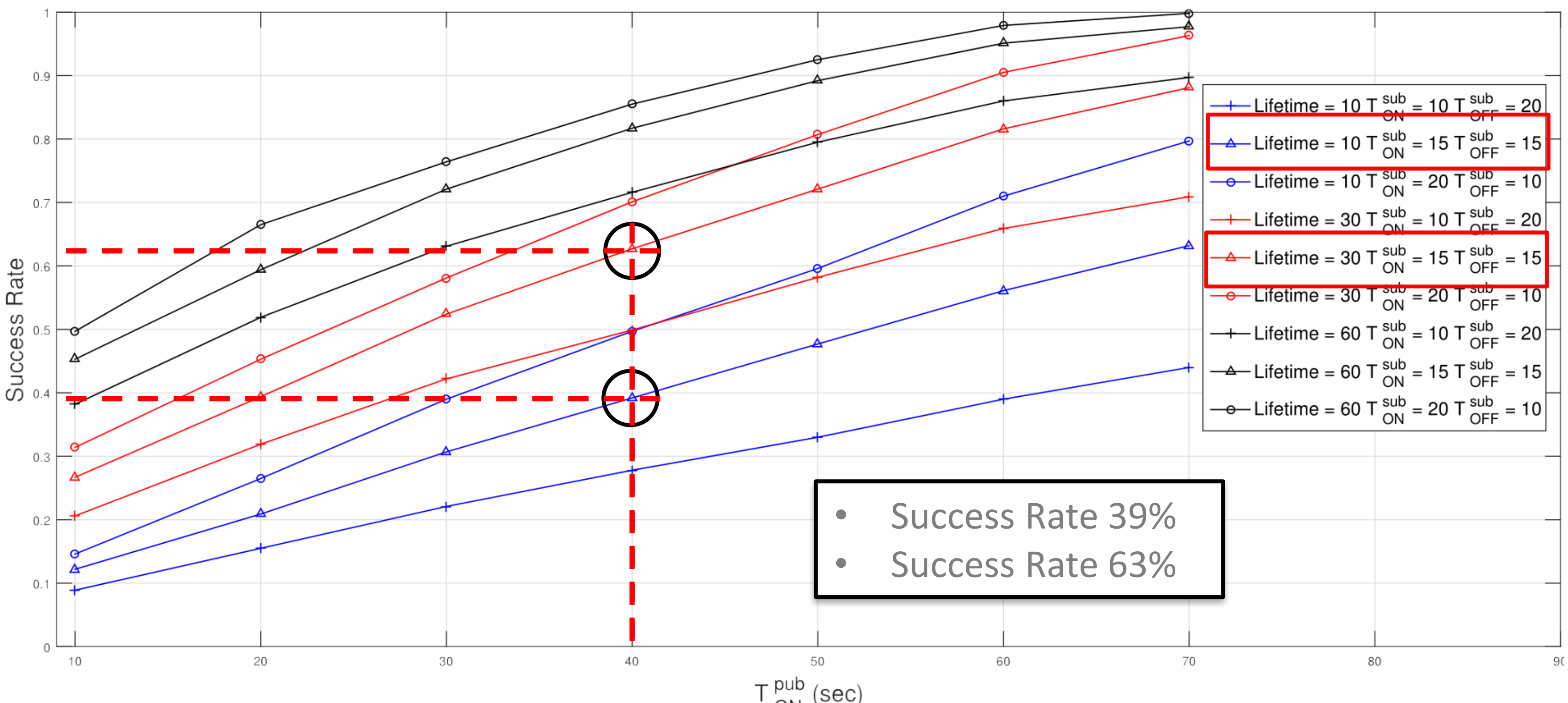
$T_{ON}^{pub} + T_{OFF}^{pub} = 80$ sec

$\lambda_{app}^{in} = 2$ events/sec

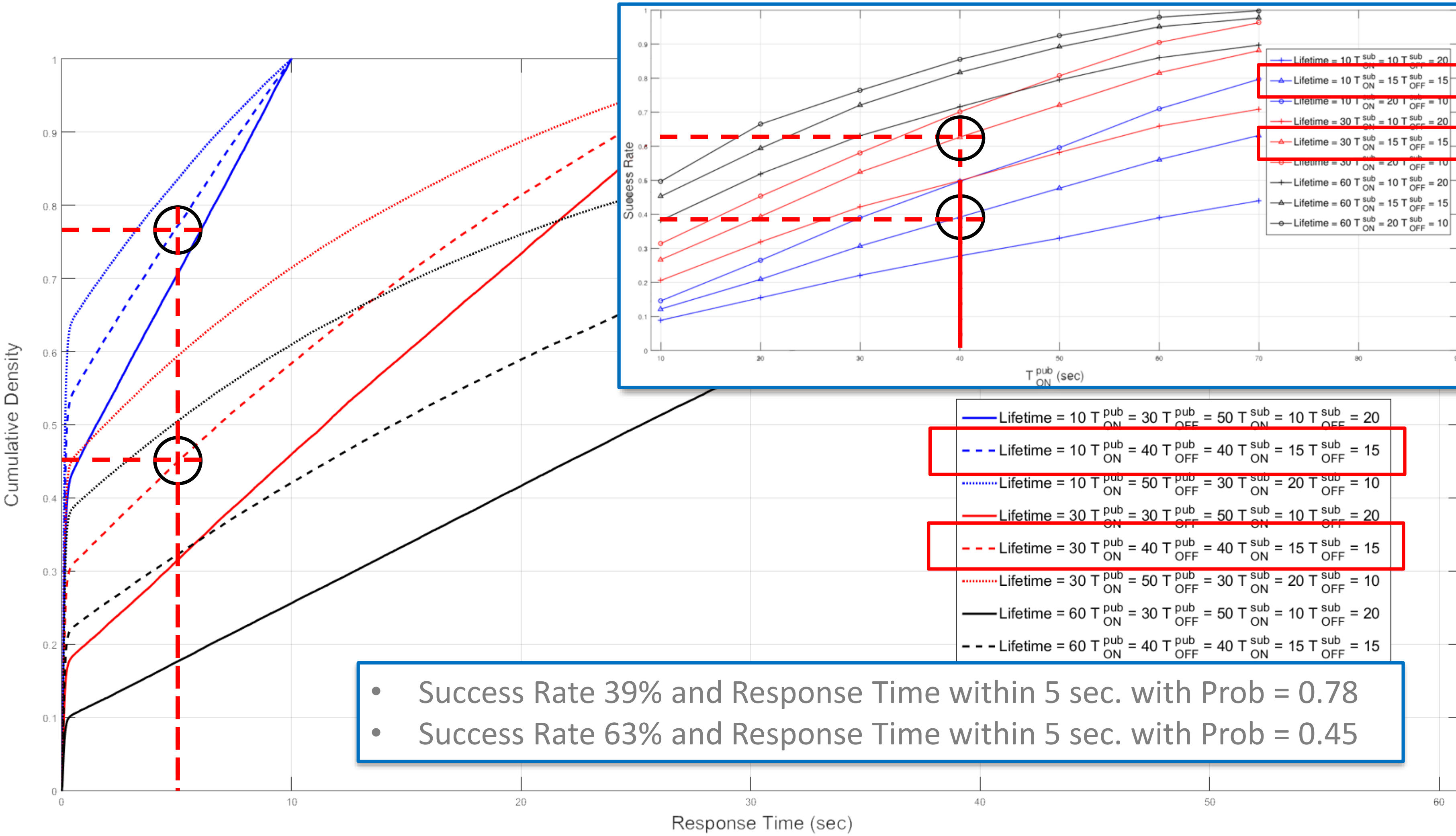
QoS_r → QoS_r

lifetime = 10, 20 and 30 sec

$T_{ON}^{sub} + T_{OFF}^{sub} = 30$ sec



p reliable to *s* reliable: response times



- Success Rate 39% and Response Time within 5 sec. with Prob = 0.78
- Success Rate 63% and Response Time within 5 sec. with Prob = 0.45

➢ Lower lifetime periods produce improved response time (but with lower success rates)

p reliable to *s* unreliable

$T_{ON}^{pub} + T_{OFF}^{pub}$ = 80 sec

λ_{app}^{in} = 2 events/sec

QoS_r → QoS_u

1. lifetime = 10sec
2. lifetime = 30sec

$T_{ON}^{sub} + T_{OFF}^{sub}$ = 30 sec

1. Success Rate 30% and Response Time within 10 sec. with Prob = 1
2. Success Rate 38% and Response Time within 10 sec. with Prob = 0.78

- QoS_u feature at the subscriber side provides markedly improved response times.
- success rates are severely bounded only by the subscriber's disconnections.

TIM system tuning

T_{ON}^{pub} = 50 sec T_{OFF}^{pub} = 40 sec

λ_{app}^{in} = 20 events/sec

T_{ON}^{sub} = 20 sec T_{OFF}^{sub} = 10 sec

QoS_r → QoS_u

1. lifetime = 10sec
2. lifetime = 30sec

1. Success Rate 37% and average Response Time 2.1 sec
2. Success Rate 47% and average response time 9.49 sec

➢ The designer now applies finite capacity buffers (K)

lifetime = 30sec

1. K = 10
2. K = 50

1. Success Rate 42% and average Response Time 0.43 sec
2. Success Rate 44% and average Response Time 1.8 sec

Next steps

- We introduce a performance modeling pattern (PerfMP) that captures the application and middleware layers of pub/sub protocols.
- System designers can rely on our PerfMP to evaluate the performance of multiple IoT applications running over pub/sub.

- Future work
  - Introduce analytical models for applications applying lifetime periods to each published event.
  - Use our analysis for system tuning at runtime.

Thank you

informatics mathematics

TATA

TATA CONSULTANCY SERVICES

Experience certainty.