Design of Cyber Bio-Analytical Physical Systems: Formal Methods, Architectures, and Multi-System Interaction Strategies

Kanwal Ashraf* kanwal.ashraf@taltech.ee
Yannick Le Moullec* yannick.lemoullec@taltech.ee
Tamás Pardy*,** tamas.pardy@taltech.ee
Toomas Rang*,** toomas.rang@taltech.ee
*Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, Estonia
**Department of Chemistry and Biotechnology, Tallinn University of Technology, Estonia

Abstract

Integration of the Cyber-Physical System (CPS) concept with bio-analytical devices is highly desirable to enable device automation as well as to improve diagnostic and analytical capabilities. However, the modeling must account for entity interactions, system dynamics, and non-functional aspects required for proper device functionality.

This paper presents a model-based system architecture that builds upon an extended timed automata-based formal technique. In contrast to prior works that utilized SysML or UML-based models, this allows for the wireless control of bio-analytical instruments. Using this formal method, the UPPAAL tool is used to model and test a case study called "A droplet flow cytometer for testing bacteria's susceptibility to antibiotics." The study shows the implications of formal techniques for the design and verification of wireless automation of high-throughput laboratory setups in Model-Based System Engineering (MBSE). Moreover, the paper extends the above aspects by adding the possibility to model multi-system interaction. This is used to analyze the trade-off between centralized and decentralized information flow strategies for better system performance under delay and bandwidth constraints. UPPAAL Stratego is used to analyze strategies for achieving specific delays and bandwidth consumption while avoiding packet losses in the event of network congestion. The results show that under strict delay constraints and high traffic, the use-case system selects the decentralized strategy over the centralized strategy. In low-traffic scenarios, the centralized strategy is more effective at ensuring the reliable operation of systems.

Keywords: Cyber Bio-analytical Physical Systems (CBPSs) Wireless Networked Control Systems Formal Methods Formal Verification Biochemical analysis Biological system modeling.

Authors' version (preprint) of the paper accepted in Microprocessors and Microsystems (2023), DOI: https://doi.org/10.1016/j.micpro.2023.104780

1 Introduction

Due to the global spread of various epidemics, there has been a significant increase in the demand for automation in laboratory-based biochemical analysis and handheld rapid diagnostic devices over the past few years. The primary factors driving this increased demand are the need to advance drug development, analysis, testing, and rapid diagnostic methods [1, 2, 3]. However, there are several challenges associated with achieving this automation, and the problem complexity is well-defined in our previous works [4, 5] i.e. automation of chip design, sensor technology, light source, and fabrication processes.

To support automation in bioanalytics, the integration of Cyber-Physical System (CPS) concepts with the microfluidic and biochemical domains has become very topical. CPSs integrate physical and computational processing via a communication network [6, 7] and account for user feedback via the cloud. Besides, feedback is also provided from the control system to the physical processes over the communication channel and vice versa [8]; this feedback is important to obtain a robust and reliable performance of the system under dynamic changes.

When designing a CPS, it is desirable to model the system for analyzing its behavior at the early design stage because the systems are usually complex. The essential points that should be considered while designing CPSs include a range of functional and non-functional properties. Functional properties depict a particular behavior of the system, while non-functional properties include time-related properties, physical properties, and behavioral properties. The reason to take these properties into account is to ensure the safe automated operation of the system under practical constraints.

Several architectures and models that integrate CPS concepts with medical devices have been proposed in recent years to enable automation in the healthcare field. The integration of CPS concepts with bioanalytical devices is still new and under-researched, and there are very few works proposing an architecture and model for the design of these devices.

In [9], the authors proposed a modified schematic for bio-analytical devices by separating the fluidic and biochemical domains from physical processes in the CPSs architecture. Another research [10] proposed an evolvable system architecture for lab-on-a-chip (LOC) devices. That work was extended to define platform-based design for LOC devices [11] and later a model-based system design framework was proposed for life sciences instruments [12]. The works [9, 12] motivated model-based system design methodology in bio-analytical and diagnostic devices and modeled the system using SysML. It is possible to model system behavior using SysML; however, as mentioned earlier in this paper, non-functional properties are required for the synchronized operation of the system under constraints; and cannot be fully handled with this approach. Furthermore, as the number of interconnected systems increases, so do the variables used to account for this interaction, necessitating a more simplified and reliable modeling approach.

Moreover, neither of the aforementioned works focused on the communication domain of the device in detail. The introduction of wireless control in bioanalytical devices could enable several features, including scalability, robustness, and remote operation of these devices, but will also introduce significant challenges. According to the authors' knowledge, despite its many features and advantages, wireless

control of bioanalytical devices has not been investigated in depth.
This paper seeks to address this research gap by providing a wireless control model for bioanalytical devices. Figure 1 depicts a simplified architecture concept for wireless control of a bio-analytical CPS, also known as CBPS [12] and CPBS [9] by introducing the communication domain, in addition to the physical, fluidic, and cyber domains. The cyber domain handles process/system control, computation, decision-making, and data storage and discarding. The communication domain connects the cyber domain with the physical and fluidic domains and is responsible for network management, information flow, and supervision. The physical and fluidic domains deal with bio-chemical reactions and physical processes.

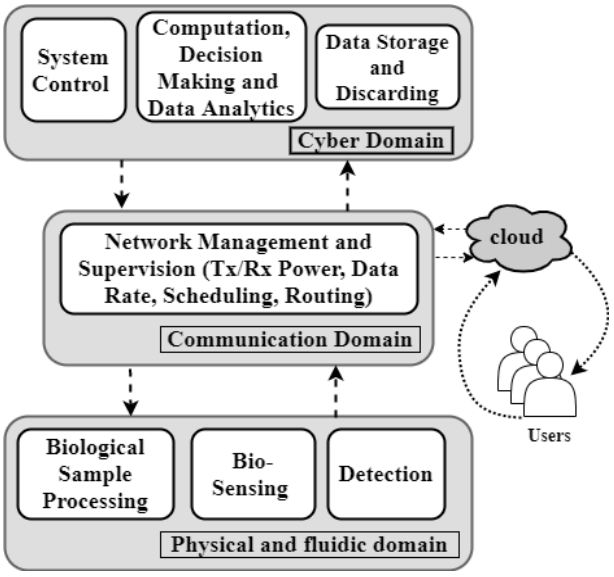

Figure 1: Bio-analytical CPS concept with cyber, communication, and physical and fluidic domains. The cyber domain is responsible for system control, computation, decision-making, as well as data storage and discarding. Next, the communication domain connects the cyber domain with the physical and fluidic domain; it performs network management, information flow, and supervision. Finally, the physical and fluidic domain deals with the biological sample processing, biosensing, and detection aspects.

So far, the best method for modeling the non-functional properties of CPSs is to use formal techniques. Formal techniques use a mathematical approach to precisely define system behavior [13]. Some of the formal methods are Pi-calculus, Petri-net, and Hybrid automata. Because of their ability to model both the continuous and discrete behavior of the system, hybrid automata [14, 15] are the most commonly used formal technique in CPSs modeling. Timed automata, a sub-class of hybrid automata, are widely used for modeling a wide range of real-time systems as they can completely automate the verification and validation process, which is our main motivation.
This paper is an extended version of [16] and provides a more extensive review of the state of the art and proposes additional features and corresponding results related to the modeling of multi-system interaction and analysis of the trade-off between centralized and decentralized

information flow strategies. The major contributions of this paper are as follows:

- Using extended timed automata, we propose a novel model-based system architecture concept for event-triggered wireless control of bio-analytical CPSs.
- We specify and verify the proposed system concept using extended timed-automata in UPPAAL [17], for a droplet flow cytometer for antibiotic susceptibility testing of bacteria in a case study.
- To analyze the interaction of several devices, we evaluated the system's trade-off using UPPAAL to choose between centralized and decentralized communication architectures under known and unknown traffic patterns.

An investigation of the system's software and hardware requirements for a baseline practical implementation of the use-case was made. For the system to be resource-efficient, event-triggered wireless control must be implemented. Modeling the system as event-triggered rather than time-triggered is primarily motivated by the fact that biological processes are often slow and can take minutes to hours to complete, whereas communication occurs in the order of milliseconds. The non-functional properties, specifically the time-dependent occurrence of different events in terms of synchronization for the whole system, are ensured. Furthermore, using stochastic timed automata, the high-level interaction between two systems for delay and resource constraints is investigated.

To the best of the authors' knowledge, this is the first work that models a bio-analytical system using a formal method, takes wireless network constraints into account, and investigates the high-level interaction of systems. The rest of the paper is organized as follows: Section 2 highlights related work in this field; Section 3 describes the preliminaries for the modeling framework and discusses the problem formulation. The timed-automata-based modeling for the case study is presented in Section 4. Model verification and a multi-system scenario for optimal system performance under constraints are covered in Section 5. Section 6 summarizes the software and hardware implementation considerations, and Section 7 concludes the paper and provides some suggestions for future directions.

2 Related Work

CPSs have applications in several disciplines, including healthcare, education, smart grids, the automotive industry, etc. [18]. In [13], a comprehensive survey highlighting different modeling languages and techniques and what aspects each of them lacks has been provided. In addition to highlighting modeling requirements, the work discusses the importance of addressing functional and non-functional properties of CPS to achieve correct operation of the device. The study [19] demonstrated the importance of a simulation-based model for clinical applications when compared to practical tests. The simulated and validated models included component-based models for medical devices as well as regression models for vital sign simulation.

Medical CPS design issues and requirements are addressed in the research [20] for the telemonitoring of high-risk pregnancies use-case using a plug-and-play architecture [21] for interaction between patients and caregivers. However, that work focuses on a high-level interface model and does not address the non-functional require-

ments of the systems. The work [22] provides a comprehensive survey for computation, communication, and memory resource efficiency in CPSs. In work [23], a rule-based simulation of biochemical processes is addressed. In research [24], a service-oriented industrial CPS with cloud infrastructure support is presented. The paper [25] provided an overview of "Safe Cooperating Cyber-Physical Systems Using Wireless Communication (SafeCOP)," which focused on safety-related Cooperating CPSs (CO-CPS) characterized by wireless communication usage and uncertain operating environments.

The work [26] provides an in-depth overview of formal methods in specification, design, and verification for software and hardware systems. The majority of modeling work for bioanalytical devices is either UML or SysML based, which fails to account for non-functional system aspects. In work [27], a SysML-based model of a mobile phone-based healthcare diagnosis system was presented. As a SysML-based model is incapable of addressing non-functional system aspects, this work emphasized the significance of formal approaches. The work [9] also uses SysML-based modeling for bioanalytical devices.

The importance of functional and non-functional properties in design verification has been highlighted in other works. The paper [28] highlights multi-dimensional hardware design verification using machine learning, where the intersection of functional and non-functional properties of electronic design is analyzed. The works [29, 30, 31] present solutions for fault tolerance, safety, and reliability in industrial and automotive CPSs.

When compared to the current state of the art, our work i) models a bio-analytical system using a formal method, ii) considers the constraints of wireless networks, iii) explores the interplay between systems at a high level, and iv) brings a special emphasis on the temporal aspects of CPS.

3 Modeling Framework Basics & Problem Formulation

Timed automata [32] are a sub-class of hybrid automata with a finite number of real-valued clocks that can be reset. Timed automata can be used to describe both functional and non-functional characteristics of real-time systems, including timing behavior. In contrast to simple finite state machines, timed automata have time constraints [33]. Timed automata-based system model verification could help ensure that the system never reaches an undesirable state [34]. As discussed earlier, the formal approach adopted in this work is based on timed automata. This section provides a few essential preliminary definitions of hybrid and timed automata and their parallel composition in Table 1.

The semantics of both hybrid and timed automata are based on two rules, i.e. discrete rule for discrete state transitions and continuous time rule based on continuous time steps. Whereas for parallel compositions for $i = 1, ..., n$ systems, the discrete transitions between the edges for two systems with automata T_i and T_{i+1} is given by *Rule Synchronization* and *Rule* $Non - Synchronization$.

To obtain an event-triggered control for bio-analytical CPSs over a non-deterministic wireless network the extension of timed-automata is needed which is discussed in detail in the next section.

Automata	Definition	Rules
Hybrid Automata	**Definition 1** A hybrid automaton is a tuple H= *(L, Var, g, Γ, Edge, Act, Inv, Init)* where: $L \rightarrow$ *Set of locations, States* $\Sigma = L \times V$, *V: set of all valuations v, where* $v : Var \rightarrow \mathbb{R}$ $Var \rightarrow$ *Real-Valued Variables,* $L \rightarrow 2^{Var}$ $g \rightarrow$ *Conditional/Guard function* $\Gamma \rightarrow$ *Set of Labels* $Inv \rightarrow$ *Invariant function,* $Inv(l) \subseteq V$, $l \in L$ $Act \rightarrow$ *functions consists of set of activities,* $f : \mathbb{R}_{\geq 0} \rightarrow V$ $Edge \rightarrow$ *Set of transitions,* $Edge \subseteq L \times \Gamma \times g(Var) \times 2^{Var} \times L$ $Init \rightarrow$ *Initial Location,* $Init \subseteq L$	**Discrete Rule** The discrete rule governing transitions between states is written as $(l, v) \xrightarrow{a} (l^{'}, v^{'})$ for $(l, a, (v, v^{'}), l^{'}) \in Edge$ and invariant $v^{'} \in Inv(l^{'})$ must hold. On the other hand, the time rule governing the time can pass in the current location and the variable related to the location can evolve given that $f \in Act$ such that $f(0) = v, f(t) = v^{'}$. **Continuous Time Rule** The continuous time rule is given by $(l, v) \xrightarrow{a} (l, v^{'})$ for $f[(0, t)] \in Inv(l)$ and t is strictly positive.
Timed Automata	**Definition 2** A timed automaton is a tuple $T = (L, \Gamma, Edge, C, Inv, Init)$ where: $L \rightarrow$ *Set of locations, States* $\Sigma = L \times V$, *V: set of all valuations v, where* $v : Var \rightarrow \mathbb{R}$ $C \rightarrow$ *Set of real-valued clocks* $\Gamma \rightarrow$ *Set of Labels* $Inv \rightarrow$ *function that assigns set of invariants to locations based on Clock Constraints (CC),* $L \rightarrow CC(C)$ $Edge \rightarrow$ *Set of transitions,* $Edge \subseteq L \times CC(C) \times \Gamma \times 2^{C} \times L$ $Init \rightarrow$ *Set of Initial States,* $Init \subseteq L$	The semantics for timed-automata are given for the discrete rule and continuous rule: **Discrete Rule** $$(l, v) \xrightarrow{a} (l^{'}, v^{'}) : (l, a, (g, C), l^{'}) \in Edge, v \models g, v^{'} = reset(C)\ v, v^{'} \models Inv(l^{'}) \quad (1)$$ where g is the guard. **Continuous Time Rule** $$(l, v) \xrightarrow{t} (l, v^{'}) : t \in \mathbb{R}_{\geq 0}, v^{'} \models Inv(l), v^{'} = v + t \quad (2)$$
Parallel Composition	**Definition 3** The parallel timed automata composition $T_1 \parallel T_2 ... \parallel T_n$ of n systems such that $T_1 = (L_1, \Gamma_1, Edge_1, C_1, Inv_1, Init_1), T_2 = (L_2, \Gamma_2, Edge_2, C_2, Inv_2, Init_2), ..., T_n = (L_n, \Gamma_n, Edge_n, C_n, Inv_n, Init_n)$, and such that clocks and states are pairwise disjoint, is given by: • $L = L_1 \times L_2 \times ... \times L_n$ • $C = C_1 \times C_2 ... \times C_n$ • $\Gamma = \Gamma_1 \times \Gamma_2 ... \times \Gamma_n$ • $Inv(l_1, l_2) = Inv_1(l_1) \wedge Inv_2(l_2), ...,$ $Inv(l_{n-1}, l_n) = Inv_{n-1}(l_{n-1}) \wedge Inv_n(l_n)$ for all $(l_1, l_2, ...l_n) \in L$ • $Init$=$\{((l_1, l_2, ...l_n), v) \in \Sigma | (l_1, v) \in Init_1 \wedge (l_2, v) \in Init_2, ... \wedge (l_n, v) \in Init_n\}$	**Rule Synch** : $$\frac{(l_i, a, (g_i, C_i), l_i^{'}) \in Edge_i, (l_{i+1}, a, (g_{i+1}, C_{i+1}), l_{i+1}^{'}) \in Edge_{i+1}}{((l_i, l_{i+1}), a, (g_i \wedge g_{i+1}, C_i \times C_{i+1}), (l_i^{'}, l_{i+1}^{'})) \in Edge} \quad (3)$$ **Rule Non − Synch$_i$** : $$\frac{(l_i, a, (g, C), l_i^{'}) \in Edge_i, a \notin \Gamma_{i+1}}{((l_i, l_{i+1}), a, (g, C), (l_i^{'}, l_{i+1})) \in Edge_i} \quad (4)$$ **Rule Non − Synch$_{i+1}$** : $$\frac{(l_{i+1}, a, (g, C), l_{i+1}^{'}) \in Edge_{i+1}, a \notin \Gamma_i}{((l_i, l_{i+1}), a, (g, C), (l_i, l_{i+1}^{'})) \in Edge} \quad (5)$$
Extended Timed Automata	**Definition 4** Extended timed automata is a tuple $T = (L,$ $\Gamma, Edge, C, Var, Chan, Inv, Init, P)$ Where $L \rightarrow$ *Set of locations, States* $\Sigma = Loc \times V$, *V: set of all valuations v, where* $v : Var \rightarrow \mathbb{R}$ $C \rightarrow$ *Set of real-valued clocks* $Var \rightarrow$ *Set of non-clock real-valued local variables* $Chan \rightarrow$ *Set of non-clock shared variables* $\Gamma \rightarrow$ *Set of Labels* $Inv \rightarrow$ *function that assigns a set of invariants to locations based on Clock Constraints (CC), local variables constraints* $\Phi(Var)$ *and set of shared channel variables* $\varphi(Chan)$, $L \rightarrow CC(C) \wedge \Phi(Var) \wedge \varphi(Chan)$ $Edge \rightarrow$ *Set of transitions,* $Edge \subseteq L \times CC(C) \times \Phi(Var) \times \varphi(Chan) \times \Gamma \times 2^{C} \times 2^{Var} \times 2^{Chan} \times L$ $Init \rightarrow$ *Set of Initial States,* $Init \subseteq L$ $P \rightarrow$ *Assigns user−defined exponential delay rate (e) to each location,* $L \rightarrow e(P)$, *The exponential rate P defines an exponential distribution to leave each state under unbounded delays.*	Extended Timed Automata follow the same rules as Timed Automata's discrete and continuous rules.

Table 1: Definitions and Rules for Hybrid Automata, Timed Automata, Parallel Composition, and Extended Timed Automata

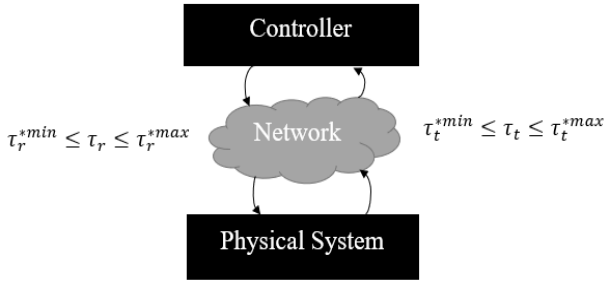

Figure 2: Wireless control over a non-deterministic network, where τ_t^{*min}, τ_r^{*min}, τ_t^{*max}, τ_r^{*max} are the minimum and maximum times for a sample to be transmitted and received, respectively.

3.1 Event-Triggered Control system with Non-deterministic Network

Assuming that each bio-analytical system acts as a Linear Time Invariant (LTI) system, we have the following equation:

$$\dot{\zeta}(t) = A\zeta(t) + Bx(t). \tag{6}$$

where $\zeta(t)$ is the current state of the system, $\dot{\zeta}(t)$ is the next state of the system and $x(t)$ is the feedback for the system, depending upon controller gain K. Assuming the network is non-deterministic, i.e. the time taken by each sample to reach its destination is unknown, the delay associated with each sample to be transferred is τ_t and for reception is τ_r. Moreover, there exists minimum times τ_t^{*min}, τ_r^{*min} and maximum times required τ_t^{*max}, τ_r^{*max} for the sample to be transferred and received, respectively as depicted in Figure 2. For a sequence of communication events τ_{ti} and τ_{ri} where $i \in \mathbb{N}$, the following assumption can be made:

$$\tau_t^{*min} \leq \tau_{ti+1} - \tau_{ti} \leq \tau_t^{*max}, \quad 0 < \tau_t^{*min} \leq \tau_t^{*max} \tag{7}$$

$$\tau_r^{*min} \leq \tau_{ri+1} - \tau_{ri} \leq \tau_r^{*max}, \quad 0 < \tau_r^{*min} \leq \tau_r^{*max} \tag{8}$$

To account for the network delays in more detail, one can refer to [35]. The feedback provided by the system to the controller will suffer delay based on the network delay [36]. Equation 9 reflects the controller feedback accounting for network delay.

$$v(t) = K\zeta(t) \quad t \in [t_k + \Delta, t_{k+1} + \Delta] \tag{9}$$

where Δ depends upon τ_r and τ_t. Hence, the difference between the sampled state and the current state is given by equation 10.

$$e(t) = \zeta(t_k) - \zeta(t) \quad t \in [t_k + \Delta, t_{k+1} + \Delta] \tag{10}$$

Employing the event-triggered approach based on the same principle as mentioned in [37], the sampled trigger time rule is given by:

$$t_{k+1} = min\left\{t | t > t_k \quad : \quad |e(t)|^2 \geq \sigma |\zeta(t)|^2\right\} \tag{11}$$

For sampled state x the inter-sample time is given by equation 12 based on trigger coefficient σ

$$\tau_\sigma = min\left\{t | |e(t)|^2 \geq \sigma |\zeta(t)|^2 \quad : \quad \zeta(0) = x\right\} \tag{12}$$

3.2 Timed Automata based Model for Non-Deterministic Wireless Communication

For the bio-analytical devices consisting of various subsystems, the proposed approach is to control each sub-unit using an event-triggered mechanism, while the network scheduling is priority based. For a simple use case, let σ_i be the trigger coefficient defined by event-triggered system control over non-deterministic channel for the i^{th} single sub-system in the whole system, where

$\sigma_i \in [0, \bar{\sigma}]$. The clock constraints (or guards) related to this sub-system are given by

$$\tau_s^{\sigma_i} \leq c \leq \tau_s^{\bar{\sigma}_i} \quad . \tag{13}$$

Assuming that the subsystem has only two states, i.e. Init and Process, the timed automata model of the sub-system can be written as:

- $L = \{Init, Process\}$
- $C = \{c\}$
- $\Gamma = \{a, b\}$
- $Edge = \{(Init, a, (c \geq \tau_s^{\sigma_i}, \{c\}), Process), (Process, b, (c \geq \tau_s^{\bar{\sigma}_i}, \{c\}), Init)\}$
- $Inv = \{Inv(Init) : c \leq \tau_s^{\sigma_i}, Inv(Process) : \tau_s^{\sigma_i} < c < \tau_s^{\bar{\sigma}_i}\}$
- $Init = \{(Init, v_0)$ with $v_0(c) = 0\}$

To include non-clock local and shared variables (channels) and to model stochastic network behavior, we have extended the timed automata as in Table 1.
The extended model is well-suited to control parameters like volume, flow-rate and synchronization between the bio-analytical devices.

4 Case-Study: Bio-Analytical Devices

The basic working principle of a droplet-based flow cytometer for the analysis of bacterial antibiotic susceptibility is based on droplet generation on a large scale, where each droplet encapsulates a single cell or a small population of bacteria, reagents, and antibiotics [1]. The droplets are then incubated to allow bacteria to grow or die based on their resistance to the specific antibiotic concentration. After the incubation, several images are captured using a high-speed camera. The camera images are then classified using a Machine Learning (ML) algorithm where dead and alive cells are identified. Depending on the ratio of dead to live cells, the bacterial susceptibility to a specific concentration of the antibiotic can be determined. Figure 3 shows the major sub-blocks of the considered droplet flow cytometer for the analysis of the antibiotic susceptibility of bacteria.

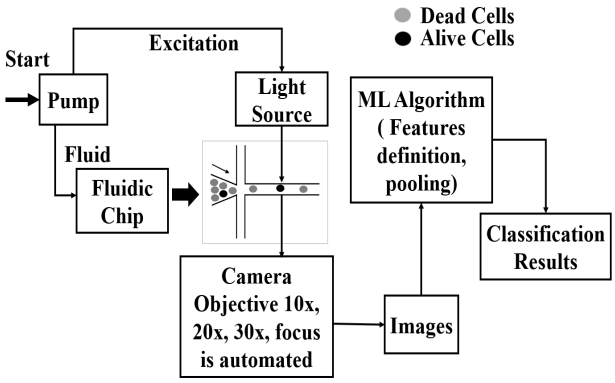

Figure 3: Use case: flow cytometer for antibiotic susceptibility of bacteria, where the main building blocks are the droplet generation unit (pump, fluidic chip), imaging unit/sensing unit (light source, camera), and detection unit (ML algorithm, classification results.

Based on the major building blocks of the droplet flow cytometer, the system can be divided into three major units, i.e., the Droplet Generation Unit, Imaging Unit/Sensing Unit and Detection Unit. Each of the sub-blocks has a control unit that is connected to the central controller over a wireless communication network. A brief description of each sub-unit and wireless control unit is given below.

4.1 Droplet-Generation Unit

The droplet generation unit consists of a pump for controlling the volume of pre-processed (incubated) fluids at a specific rate for a specified duration and a microfluidic chip. The droplet generation starts with an initialization command to the pump with a defined flow rate and duration. The central controller communicates over the network for initialization and goes into sleep mode until it is triggered by the sub-unit with an acknowledgment of its task completion.

4.2 Imaging Unit

The imaging unit consists of a light source and a high-speed smart camera unit that is capable of capturing images at a variable frame rate (FR). The imaging unit is also able to adjust the resolution and Depth of Field (DoF). The control of the imaging unit activates via a central controller at a specified FR, resolution, and DoF, which are adjustable by the smart camera unit depending on the required image quality.

4.3 Detection Unit

The detection unit consists of a microcontroller able to classify the images using machine learning. The used ML algorithm could be based on deep learning using artificial neural networks, where a trained classifier can classify images depending upon their features.

4.4 Wireless Control Unit

Upon instantiating, the central controller activates each sub-unit in a specified order over a wireless communication network and goes to sleep or low power mode unless a trigger command is received to acknowledge the task completion by the sub-units. As mentioned earlier in Section 3.2 the network scheduling is priority-based, and there is a specified duration for a non-deterministic network to receive and send the commands, both to the controller as well as to the sub-units. The paths or run-time transitions for the modeled wireless control of these devices are based on boolean variables and channels, as well as clock variables. They are given as:

Controller Run Time Transitions

$$Initialization \xrightarrow{\text{d1:U1,Chan}} Droplet\_Generation$$
$$\xrightarrow{\text{y:Init2,C}}: Init\_U2 \xrightarrow{\text{d2:U2,Chan}} Imaging, ...$$
$$Detection \xrightarrow{\text{y:finish,C}} Initialization$$

with invariants as below:

$$Inv = \{Inv(Droplet\_Generation) : y{<}tc1\_max,$$
$$Inv(Imaging) : y{<}tc2\_max,$$
$$Inv(Detection) : y{<}tc3\_max\}$$

Network Run Time Transitions

$$Init \xrightarrow{\text{d1:C1,Chan}} DG\_sending$$
$$\xrightarrow{\text{z:S1,C}}: wait\_DG \xrightarrow{\text{End\_1:R1,Chan}} ACK\_DG, ...$$
$$ACK\_DT \xrightarrow{\text{Rx3:end,Chan}} Init$$

with invariants as below:

$$Inv = \{Inv(DG\_sending) : z{<}tx1\_max,$$
$$Inv(wait\_DG) : z{<}rx1\_max, Inv(ACK\_DG) : z{<}$$
$$rx1\_max, Inv(Im\_Sending) : z{<}tx2\_maxInv(wait\_Im) :$$
$$z{<}rx2\_max, Inv(ACK\_Im) : z{<}rx2\_max, Inv(DT\_$$
$$Sending) : z{<}tx3\_maxInv(wait\_DT) : z{<}rx3\_max,$$
$$Inv(ACK\_DT) : z{<}rx3\_max\}$$

The run time transitions for other sub-units and their parallel composition can be formulated based on the same principles as mentioned here and in Section 3.

5 Verification with UPPAAL & Multi-System Interaction

Model verification is required to ensure that the designed system meets all specifications, such as time constraints, synchronization, and the absence of deadlocks. For model checking and verification, we used UPPAAL. UPPAAL is a model checking, verification, and validation tool [17]. During verification, the system was analyzed under bounded, unbounded, and probabilistic delay distributions using Stochastic Model Checking (SMC). Figures 4 and 5 show the control and network models executed in UPPAAL for generic subsystems with no specific functionality defined. Here, each of the states has an upper bound on the time to stay in a specific state; as for "Droplet Generation state" this value is bounded by $tc1\_max$ for "Imaging" state it is bounded by $tc2\_max$ and for "Detection state" it is bounded by $tc3\_max$. Each of the sub-systems was then further modeled in depth. For conciseness, the other sub-system models can be found at : UPPAAL Models. In the network unit (Figure 5), the state transition is also bounded by defined transmission variables (guards) such as $tx1\_max$ ($DG\_sending$), $tx2\_max$ ($Im\_Sending$) and $tx3\_max$ ($DT\_Sending$). A detailed description of other variables used in the network and control unit is provided in Table 2.

In addition to the use of the UPPAAL simulator, the models have been further verified using different queries based on the UPPAAL lan-

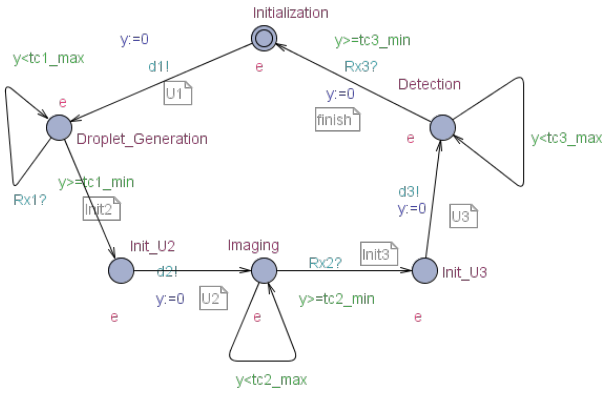

Figure 4: Control unit for droplet flow cytometer. The major states are: Initialization, Droplet Generation, Detection and Imaging with upper bound on time transition from state to state: $tc1\_max$ (Droplet Generation state), $tc2\_max$ (Imaging), $tc3\_max$ (Detection).

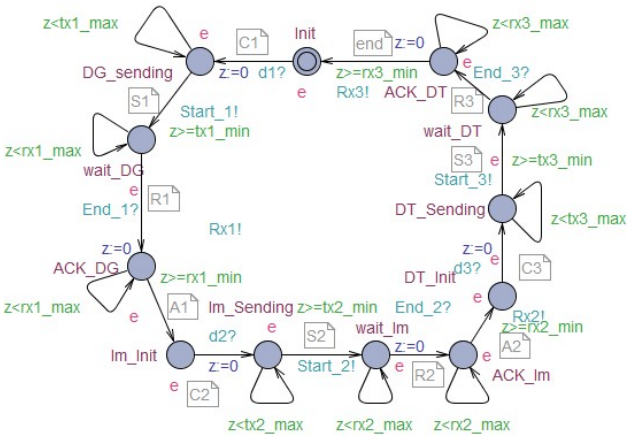

Figure 5: Network unit for droplet flow cytometer, where the main states are: $DG\_/Im\_/DT\_sending$, $wait\_DG/\_Im/\_DT$, and $ACK\_DG/\_Im/\_DT$.

Table 2: Description for variables in UPPAAL Models

Description	***Variables***
Upper bound for time transition for different states in control unit	$tc1\_max$ $(Droplet\_Generation \rightarrow Initialization)$, $tc2\_max$ $(Init\_U2 \rightarrow Imaging)$, $tc3\_max$ $(Init\_U3 \rightarrow Detection)$
Lower bound for time transition for control unit	$tc1\_min(Droplet\_Generation \rightarrow Init\_U2)$, $tc2\_min(Imaging \rightarrow Init\_U3)$, $tc3\_min$ $(Detection \rightarrow Initialization)$
Transmission time upper bounds for network unit	$tx1\_max$, $tx2\_max$, $tx3\_max$
Reception time upper bounds for network unit	$rx1\_max$, $rx2\_max$, $rx3\_max$
Information Transmission states in network unit	$DG\_sending$,$Im\_Sending$, $DT\_Sending$
Initialization states in network unit	$Init$, Im_init, DT_Init
Acknowledgment states in network unit	ACK_DG, ACK_Im, ACK_DT
Wait states in network unit	$wait\_DG$, $wait\_Im$, $wait\_DT$
Channel synchronization variables in control unit	$d1$, $Rx1$, $d2$, $Rx2$, $d3$, $Rx3$
Channel synchronization variables in network unit	$d1$, $Start\_1$, End_1, $Rx1$, $d2$, $Start\_2$, End_2, $Rx2$,$Start\_3$, End_3,$Rx3$

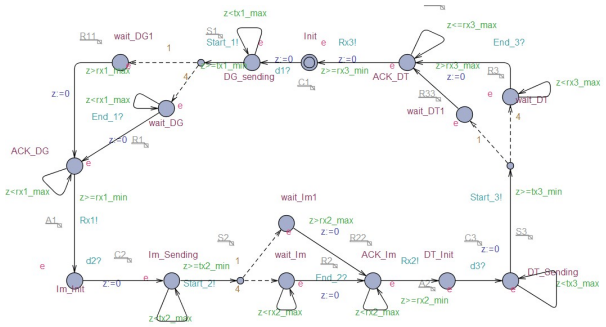

Figure 6: Network unit with discrete probabilistic non-delayed and delayed paths (1/4, 4/5)

Table 3: Verification Queries

Queries	***Properties***
E <> $Communication.wait$ and $z \geq tx1\_max$	Delay
A[]!($Communication.Sending$&& $Bio\_Chip\_control.Rate\_Definition$)	Synchronization
A[] $Communication.wait$ imply$z \geq$ tx1_min	Reset
simulate[< = 300]Network.ACK_DG, Control. $Droplet\_Generation$, $Imaging.Process\_D2$	Simulation
Pr[=<100](<> Network.DG_sending) > = Pr[<= 100](<> $Control.Imaging$)	Probability Comparison

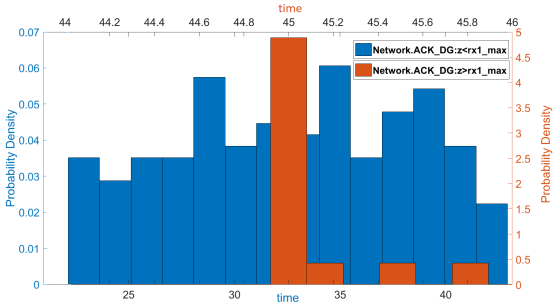

Figure 7: Probability density distribution for the network unit with acknowledgment time lower (blue bars) and higher (orange bars) than the threshold ($rx1\_max$)

guage reference guide. Examples of some of the queries used for system verification are given in Table 3. Using the query-based verification system, state transitions, probability density distributions, and probability comparisons of states at different intervals could be verified.

In addition to an exponential delay distribution, the network model could include discrete probabilistic delays for different paths in the model.

Figure 6 shows an example of the network

model with discrete probabilities defined for delayed and non-delayed paths. Figure 7 shows the probability density distribution for ACK_DG state for a time lower than (blue bars) and higher than (orange bars) $rx1\_max$ threshold, with confidence 0.95, for the model depicted in Figure 6. Here the discrete probabilistic choices (1/5, 4/5) associated with the paths increase the probability density associated with $z > rx1\_max$. The discrete probabilistic choice for each path could help in the estimation of the delay in any system. Additionally, a cost and reachability time analysis could also be performed for each path. The formal verification showed that the modeled use case didn't break any time limits and that the operation of the device happened in the right order and at the same time.

So far, we have considered the problem for a single device with sub-units; however, CBPS are real-time distributed systems by nature, with multiple devices interacting with each other in both competitive and cooperative ways [38, 39] to achieve better performance. In this section, the single-use case problem is extended to a multi-system problem for optimization of delay and bandwidth consumption.

When dealing with multi-system interaction [40, 41] a centralized CPS control approach [42] might become inefficient as the systems might be highly distributed in space and overall computational complexity could increase drastically. Therefore, a decentralized and distributed network control approach is more suitable. Figure 8 shows an overview of generalized centralized, decentralized, and distributed system architectures. In case of scarce network resources such as limited bandwidth or strict time-delay restrictions, a distributed or decentralized CPS architecture [43] is more desirable. Decentralized and distributed communication and control architectures are already present around us for many applications, e.g, microgrids [44, 45]. However, there is always a trade-off when choosing between a centralized control approach and a decentralized or distributed control approach. If the number of systems interacting with each other is less than the upper bound on network constraints, such as bandwidth and power, a centralized control and communication approach would be preferable to a distributed one, where information flow could suffer from large delays.

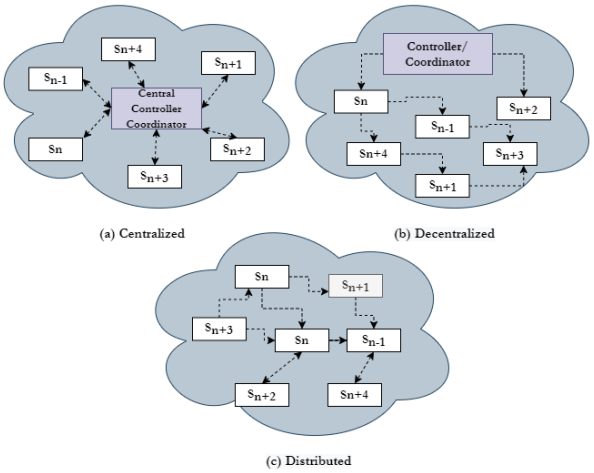

Figure 8: (a) Centralized, (b) Decentralized, (c) Distributed System Architecture for information control and transmission.

We aimed to study the necessary interaction between different devices in combination with their interaction with the sub-units of the device. A strategic approach is employed to determine whether to select a centralized or decentralized control approach, which eventually affects the overall performance of the system when traffic patterns are known or unknown. To analyze which approach is better suited, a Stochastic Timed Automata (STA)

based approach was used, where the choice of transferring information from one system to another is kept random. The definition of a stochastic timed automaton is given as:

STA:Definition: A stochastic timed automata is a tuple $STA= (TA, \rho,\omega)$ where a timed automata TA is equipped with probability measure ρ and positive weights ω. [46]

where the transition between states depends not only upon the probability of transition between states but also on the waiting time or guard on the state. The transition from state s_t to s_{t+1} is dependent on wait time (t_s) and probability p_s and is written as $s_t \overset{t_s,p_s}{\rightarrow} (s_{t+1})$. We used Uppaal Stratego [47], an optimization, modeling, and strategy exploration tool for pricing strategic timed games, to simulate the strategy. For the use-case, we modeled the interaction of two systems in which information flow between the two systems could be centralized, decentralized, or network traffic aware, with the system having the option to choose between centralized and decentralized. Building upon the centralized and decentralized architecture where a coordinator is either managing all or a few nodes, one can control the information flow between different nodes (Figure 8).

In both centralized and decentralized control, network traffic is an uncontrollable parameter (Figure 9). By making the system aware of the traffic load, we can choose which strategy is best for the information flow for optimal network resource consumption under delay constraints. As depicted in Figure 9, UPPAAL Stratego computes strategies to transfer information via centralized, decentralized, and traffic-aware information flows. Information flow from system1 to

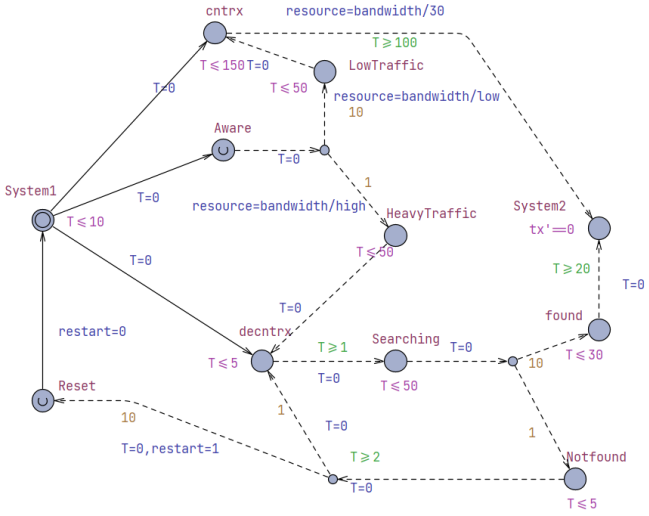

Figure 9: Centralized vs decentralized information flow control simulation using UPPAAL Stratego, where the main states are: centrx, decentrx, Aware, HeavyTraffic, and LowTraffic.

system2 may traverse controllable or uncontrollable stages (centrx, dencntrx, aware, i.e., network traffic aware). In other states, $Lowtraffic$ and $HeavyTraffic$ are associated with conditions where information passes through heavy or low traffic, as the names suggest. State $Searching$ or $found$ is associated with locating the path to the last node, $System2$, where data is transmitted.

The probability of achieving a delay below 150 ms with all the mentioned approaches is 0.76 with 95% confidence interval, whereas the average delay estimate is 120 ms with probability 0.98 with 95% confidence interval (Figure 10). The mean bandwidth consumption (i.e. achieved throughput) is 1700 kbps with probability 0.68 and 95% confidence interval (see Figure 11).

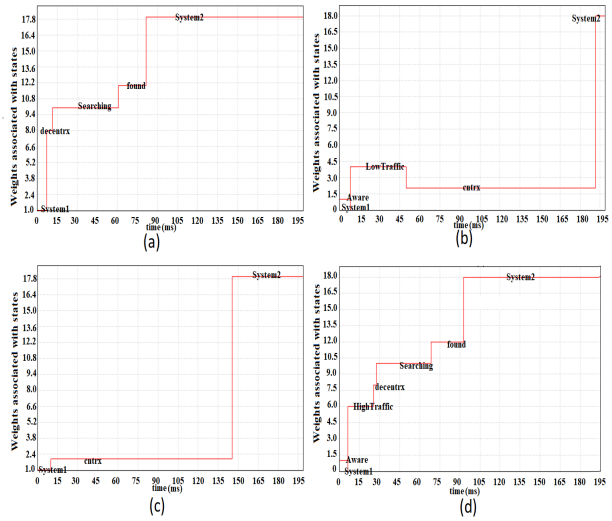

Figure 10: UPPAAL Strategies **(a):** Decentralized Information Flow (System1→decntrx→Searching→found→System2); **(b):** Traffic Aware Information Flow (System1→Aware→LowTraffic→cntrx→System2); **(c):** Centralized Information Flow (System1→cntrx→System2); **(d):** Traffic Aware Information Flow (System1→Aware→HighTraffic→decntrx →Searching→found→System2).

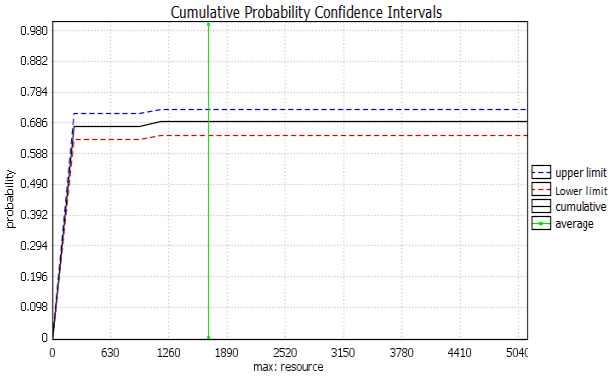

Figure 11: Bandwidth consumption: cumulative probability confidence intervals. The average bandwidth consumption is 1700 kbps as denoted by the green line.

The choice of architecture is heavily influenced by the delay constraints. Four different strategies were investigated to figure out how to choose between decentralized, centralized, and traffic-aware communication.
In the first strategy, named **Lenient Delay Strategy**, the requirement is to compromise the delay requirement to avoid high network traffic scenarios and ensure the information transfer. In the second strategy, named **Stringent Delay Strategy**, the focus is on achieving minimal transmission delay. In the third strategy, named **Opt Strategy**, the middle ground where the minimal delay is achieved regardless of communication via a centralized or decentralized architecture while avoiding high network traffic scenarios is analyzed. In the fourth strategy, named **network_res Strategy**, the goal is to consume minimum network resources, i.e., bandwidth, in combination with the lowest possible delay, regardless of the network architecture analyzed.

The first strategy was analyzed with an upper bound on delay as high as 200 ms and the possibility of avoiding large delays due to high network traffic.

To obtain an overview of the strategies employed for a stringent delay requirement (Stringent Delay Strategy), the upper bound is chosen to be 100 ms without caring about network traffic; the simulation execution, the system tends to select the decentralized communication strategy in this scenario.

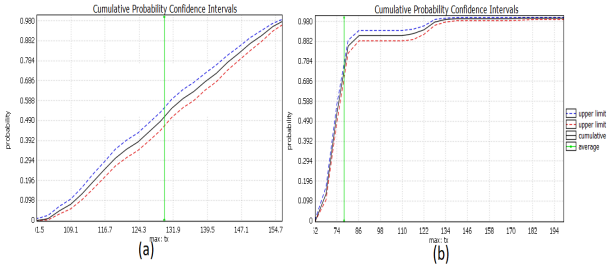

Figure 12: Strategies **(a): Lenient Strategy** Cumulative Probability Interval (Delay). Parameters: $\alpha = 0.05, \epsilon = 0.05$, Bucket Width = 1.6923, Bucket Count = 23. Mean Estimate = 129.9 ms; **(b): Stringent Delay Strategy** Cumulative Probability Interval (Delay). Parameters $\alpha = 0.05, \epsilon = 0.05$, Bucket Width = 1.6923, Bucket Count = 23. Mean Estimate = 77.94 ms.

The mean delay (Figure 12(a)) value obtained is then 129 ms with 95% confidence interval whereas the centralized communication is chosen as the main architecture. Under Stringent strategy (Figure 12(b)), the mean delay achieved is 78 ms with 95% confidence interval.

The third strategy (Opt Strategy) was employed to check the middle ground between the lowest delay achieved via decentralized or centralized strategy; the simulation results (Figure 13) show that the best-suited method for the employed use-case is to choose a centralized architecture for the guaranteed lower delay and reliable communication. The mean estimated delay is 125 ms with 95% confidence interval.

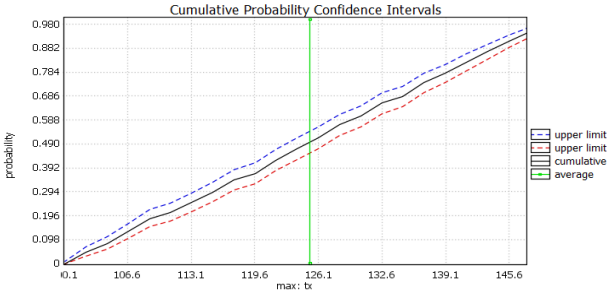

Figure 13: Strategies **Opt Strategy** Cumulative Probability Interval (Delay). Parameters: $\alpha = 0.05, \epsilon = 0.05$, Bucket Width = 1.6923, Bucket Count = 23. Mean Estimate = 125.2 ms.

The fourth strategy (network_res Strategy) is employed to minimize the delay under minimum network resource, i.e., bandwidth consumption. Figure 14 shows the cumulative probability confidence for bandwidth consumption and minimum delay. The mean estimate for bandwidth consumption is 960 kbps with a mean delay estimate of 109 ms with 95% confidence interval.

6 Implementation Considerations

The implementation of Cyber-Physical Bioanalytical devices faces critical challenges such as limited data storage and hardware computing capabilities, the need for low energy/power consumption and efficient data communication,

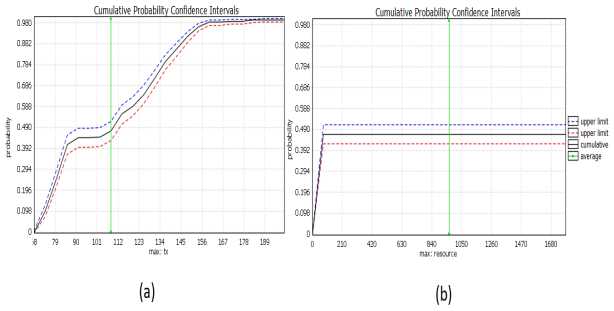

Figure 14: network_res (resource-minimization) Strategy **(a): Cumulative Probability Interval (Delay)**. Parameters: $\alpha = 0.05, \epsilon = 0.05$, Bucket Width = 1.6923, Bucket Count = 13. Mean Estimate = 109 ms; **(b) Cumulative Probability Interval (Bandwidth consumption)** Parameters: $\alpha = 0.05, \epsilon = 0.05$, Bucket Width = 1.6923, Bucket Count = 13. Mean Estimate = 960 kbps

as well as interaction with users or the environment. These critical aspects necessitate an examination of both software and hardware design aspects before the practical implementation of use cases.

6.1 Software Design Architecture

Software design of CPBS should be able to bridge the gap to enable the control of both functional and non-functional properties of the devices. While considering the software design for CPBS, the goal is to use a single publisher-subscriber interface software that enables all hardware interactions with humans and other devices to make them easy to use. To achieve the required operating capabilities in the hardware, it is necessary to take care of the following major requirements of the CPBS software:

- Information communication including remote device connection and information routing including constraint violation, errors and trigger events;
- Data handling including data storage, defining data structure, data flow control and sensing, and actuation data processing;
- Process management, scheduling, operating system capabilities, user-interface.

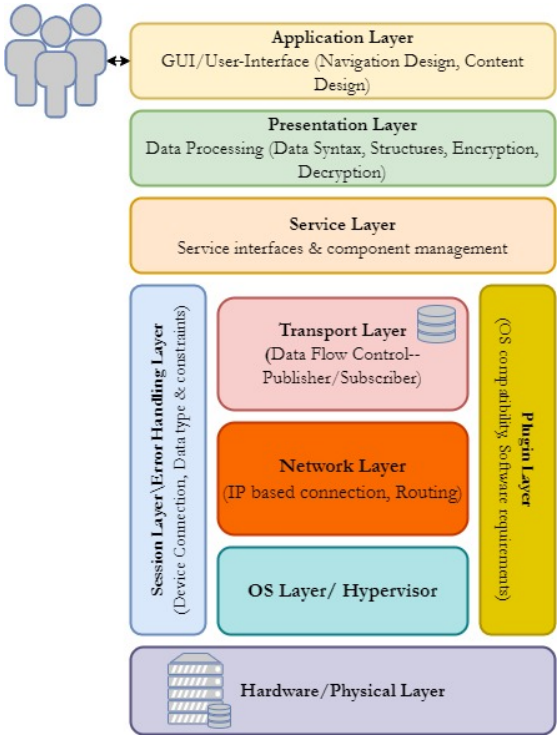

Figure 15: Proposed Layered Software Architecture for CPBS. The architecture considers the principles of service-oriented software architecture, three-tier architecture OSI model, and Cisco's OSI model for cloud computing.

Figure 15 shows the proposed multi-layer software architecture, which is designed by consid-

ering service-oriented software architecture [48, 49, 50], three-tier architecture OSI model [51] and Cisco's OSI model for cloud computing [52]. Our aim is to develop the software with capabilities such that each software service capability could be used individually for subsystems. The presentation layer allows the interface to respond to user input and user-controlled actions via Graphical User Interface (GUI) or Command-line User Interface (CUI) based-interaction. The interface is connected to several back-end processes, which include data processing or control action determination as well as communication with sub-processes and devices. The presentation layer includes data processing and lies on top of the service layer, which manages different service components and interfaces.

The plugin layer manages OS compatibility and software requirements. The session layer manages connections between different services, devices, and sub-processes. The transport layer, in combination with the network layer, handles data flow control, IP/TCP-based connections, and information routing. The low-level layers are of the same type as in any software design, e.g., hardware/physical layer and OS layer or hypervisor for management of hardware resources.

6.2 Hardware Consideration

To achieve effective control and wireless communication between different devices, the aim is to select cost-effective, reliable, and easy-to-use Single Board Computers (SBCs) as well as, where needed, simple micro-controllers. A comparison of different specifications of SBCs helped to analyze the Raspberry Pi as a strong candidate for the implementation of the communication and control algorithms. Raspberry Pi 4 is equipped with an ARM cortex, and its benchmarking showed significant performance results for image processing as well as for baseline implementation of publish-subscriber-based event-triggered wireless communication.

7 Conclusion

In this paper, an extended timed-automation-based formal technique for modeling bio-analytical CPSs with constraints over a wireless network was presented. A timed strategic approach was also used to give an overview of delay and bandwidth limits. The main goal was to provide a formal model-based architecture for bio-analytical devices to encourage the use of formal techniques in bio-analytical devices in the future. To depict the application of the proposed technique, a case study was modeled and verified using UPPAAL. It was extended to multi-system interaction using timed stochastic automata that were evaluated by means of UPPAAL Stratego. The results showed that the minimum delay achieved from system1 to system2 by a centralized communication architecture is 129 ms, 87 ms with a decentralized architecture, and 125 ms when avoiding high-traffic scenarios.

The multi-system interaction example shows how this model can be utilized in Model-based System Engineering (MBSE). Wireless connectivity will allow multiple components in a high-throughput laboratory setup to communicate without physical connections. Event-triggered control will make the system resource efficient in terms of computing and communication. The verification of both the model and strategies helped to analyze the system under bounded, unbounded, and probabilistic delay distributions.

Despite our efforts, there are still hardware and software implementation concerns for the CPBS. Non−deterministic device behavior, hardware defects, and network limits and concerns such as network failure are examples. With our present results, we are implementing the use case and analyzing it with a focus on open-source system design availability.

Acknowledgement

This research is funded by the Estonian Science Agency ETAg grant numbers PRG620, PUT1435, and TAR16013 Center of Excellence 'EXCITE IT'. This research is also supported by "TTU development program 2016-2022", project code 2014-2020.4.01.16-0032.

References

[1] O. Scheler, K. Makuch, P. Debski, M. Horka, A. Ruszczak, N. Pacocha, K. Sozański, O.-P. Smolander, W. Postek, P. Garstecki, Droplet-based digital antibiotic susceptibility screen reveals single-cell clonal heteroresistance in an isogenic bacterial population, Scientific Reports 10 (2020) 3282. doi:10.1038/s41598-020-60381-z.

[2] W. Li, H. Xi, S. H. Tan, Insights and Advancements in Microfluidics, MDPI, 2017. doi:10.3390/books978-3-03842-517-5.
URL http://www.mdpi.com/books/pdfview/book/360

[3] L. Brümmer, S. Katzenschlager, M. Gaeddert, C. Erdmann, S. Schmitz, M. Bota, M. Grilli, J. Larmann, M. Weigand, N. Pollock, A. Macé, S. Carmona, S. Ongarello, J. Sacks, C. Denkinger, Accuracy of novel antigen rapid diagnostics for sars-cov-2: A living systematic review and meta-analysis, PLOS Medicine 18 (2021) e1003735. doi:10.1371/journal.pmed.1003735.

[4] N. Gyimah, O. Scheler, T. Rang, T. Pardy, Can 3d printing bring droplet microfluidics to every lab?—a systematic review, Micromachines 12 (3) (2021). doi:10.3390/mi12030339.
URL https://www.mdpi.com/2072-666X/12/3/339

[5] K. Pärnamets, T. Pardy, A. Koel, T. Rang, O. Scheler, Y. Le Moullec, F. Afrin, Optical detection methods for high-throughput fluorescent droplet microflow cytometry, Micromachines 12 (3) (2021). doi:10.3390/mi12030345.
URL https://www.mdpi.com/2072-666X/12/3/345

[6] R. Crowder, 11 - cyber physical systems and security, in: R. Crowder (Ed.), Electric Drives and Electromechanical Systems (Second Edition), second edition Edition, Butterworth-Heinemann, 2020, pp. 271–289. doi:https://doi.org/10.1016/B978-0-08-102884-1.00011-X.

[7] K.-D. Kim, P. R. Kumar, Cyber–physical systems: A perspective at the centennial, Proceedings of the IEEE 100 (Special Centennial Issue) (2012) 1287–1308. doi:10.1109/JPROC.2012.2189792.

[8] H. Li, Chapter 1 - introduction to cyber physical systems, in: H. Li (Ed.), Communications for Control in Cyber Physical Systems, Morgan Kaufmann, Boston, 2016, pp. 1–8. doi:https://doi.org/10.1016/B978-0-12-801950-4.00001-9.

[9] E. Evin, Y. Uludağ, Bioanalytical device design with model-based systems engineering tools, IEEE Systems Journal 14 (3) (2020) 3139–3149. doi:10.1109/JSYST.2020.2993377.

[10] F. Patou, M. Dimaki, W. E. Svendsen, K. Kjægaard, J. Madsen, A smart mobile lab-on-chip-based medical diagnostics system architecture designed for evolvability, in: 2015 Euromicro Conference on Digital System Design, 2015, pp. 390–398. doi:10.1109/DSD.2015.11.

[11] F. Patou, Evolvable smartphone-based point-of-care systems for in-vitro diagnostics, 2016.

[12] F. Patou, M. Dimaki, A. Maier, W. E. Svendsen, J. Madsen, Model-based systems engineering for life-sciences instrumentation development, Systems Engineering 22 (2) (2019) 98–113. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/sys.21429, doi:https://doi.org/10.1002/sys.21429.

[13] I. Graja, S. Kallel, N. Guermouche, S. Cheikhrouhou, A. Hadj Kacem, A comprehensive survey on modeling of cyber-physical systems, Concurrency and Computation: Practice and Experience 32 (15) (2020) e4850, e4850 cpe.4850. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.4850, doi:https://doi.org/10.1002/cpe.4850.

[14] S. Geng, J. Peng, P. Li, Modeling and verification of cyber-physical systems under uncertainty, in: 2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2017, pp. 1491–1496. doi:10.1109/FSKD.2017.8392986.

[15] F. Tan, Y. Wang, Q. Wang, L. Bu, R. Zheng, N. Suri, Guaranteeing proper-temporal-embedding safety rules in wireless cps: A hybrid formal modeling approach, in: 2013 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2013, pp. 1–12. doi:10.1109/DSN.2013.6575357.

[16] K. Ashraf, Y. L. Moullec, T. Pardy, T. Rang, Model-based system architecture for event-triggered wireless control of bio-analytical devices, in: 2021 24th Euromicro Conference on Digital System Design (DSD), 2021, pp. 465–471. doi:10.1109/DSD53832.2021.00076.

[17] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, W. yi, Uppaal - a tool suite for automatic verification of real-time systems., Vol. 3, 1995, pp. 232–243. doi:10.7146/brics.v3i58.18769.

[18] H. Chen, Applications of cyber-physical system: A literature review, Journal of Industrial Integration and Management 02 (2017) 1750012. doi:10.1142/S2424862217500129.

[19] L. C. Silva, M. Perkusich, F. M. Bublitz, H. O. Almeida, A. Perkusich, A model-based architecture for testing medical cyber-physical systems, in: Proceedings of the 29th Annual ACM Symposium on Applied Computing, SAC '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 25–30. doi:10.1145/2554850.2555028.

URL https://doi.org/10.1145/2554850.2555028

[20] J. Jezewski, A. Pawlak, K. Horoba, J. Wrobel, R. Czabanski, M. Jezewski, Selected design issues of the medical cyber-physical system for telemonitoring pregnancy at home, Microprocessors and Microsystems 46 (2016) 35–43. doi:10.1016/j.micpro.2016.07.005.
URL http://dx.doi.org/10.1016/j.micpro.2016.07.005

[21] T. Li, F. Tan, Q. Wang, L. Bu, J.-N. Cao, X. Liu, From offline toward real-time: A hybrid systems model checking and cps co-design approach for medical device plug-and-play (mdpnp), in: 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems, 2012, pp. 13–22. doi:10.1109/ICCPS.2012.10.

[22] Z. Li, C. Huang, X. Dong, C. Ren, Resource-efficient cyber-physical systems design: A survey, Microprocessors and Microsystems 77 (2020). doi:10.1016/j.micpro.2020.103183.

[23] S. Ehmes, Rule-based Simulation of biochemical Reaction (2018).

[24] X. Yue, H. Cai, H. Yan, C. Zou, K. Zhou, Cloud-assisted industrial cyber-physical systems: An insight, Microprocessors and Microsystems 39 (8) (2015) 1262–1270. doi:10.1016/j.micpro.2015.08.013.
URL http://dx.doi.org/10.1016/j.micpro.2015.08.013

[25] P. Pop, D. Scholle, I. Šljivo, H. Hansson, G. Widforss, M. Rosqvist, Safe cooperating cyber-physical systems using wireless communication: The SafeCOP approach, Microprocessors and Microsystems 53 (2017) (2017) 42–50. doi:10.1016/j.micpro.2017.07.003.
URL http://dx.doi.org/10.1016/j.micpro.2017.07.003

[26] I. Rodhe, M. Karresand, Overview of formal methods in software engineering (December) (2015) 6–13.

[27] M. Dimaki, A. M. Maier, W. E. Svendsen, J. Madsen, Towards a model-based systems engineering approach to life-science instrumentation systems development, 2018.

[28] X. Lai, A. Balakrishnan, T. Lange, M. Jenihhin, T. Ghasempouri, J. Raik, D. Alexandrescu, Understanding multidimensional verification: Where functional meets non-functional, Microprocessors and Microsystems 71 (2019) 102867. doi:10.1016/j.micpro.2019.102867.
URL https://doi.org/10.1016/j.micpro.2019.102867

[29] G. Tertytchny, N. Nicolaou, M. K. Michael, Classifying network abnormalities into faults and attacks in IoT-based cyber physical systems using machine learning, Microprocessors and Microsystems 77 (2020). doi:10.1016/j.micpro.2020.103121.

[30] R. Nouacer, M. Djemal, S. Niar, G. Mouchard, N. Rapin, J. P. Gallois, P. Fiani, F. Chastrette, A. Lapitre, T. Adriano, B. Mac-Eachen, EQUITAS: A tool-chain for functional safety and reliability improvement in automotive systems, Microprocessors and Microsystems 47 (2016) 252–261. doi:10.1016/j.micpro.2016.07.020.

URL http://dx.doi.org/10.1016/j.micpro.2016.07.020

[31] S. Balasubramaniyan, S. Srinivasan, F. Buonopane, B. Subathra, J. Vain, S. Ramaswamy, Design and verification of Cyber-Physical Systems using TrueTime, evolutionary optimization and UPPAAL, Microprocessors and Microsystems 42 (2016) 37–48. doi:10.1016/j.micpro.2015.12.006. URL http://dx.doi.org/10.1016/j.micpro.2015.12.006

[32] R. Alur, D. L. Dill, A theory of timed automata, Theoretical Computer Science 126 (2) (1994) 183–235. doi:https://doi.org/10.1016/0304-3975(94)90010-8.

[33] K. G. Larsen, P. Pettersson, W. Yi, Model-checking for real-time systems, in: H. Reichel (Ed.), Fundamentals of Computation Theory, Springer Berlin Heidelberg, Berlin, Heidelberg, 1995, pp. 62–88.

[34] W. Yi, P. Pettersson, M. Daniels, Automatic Verification of Real-Time Communicating Systems by Constraint-Solving, Springer US, Boston, MA, 1995, pp. 243–258. doi:10.1007/978-0-387-34878-0_18. URL https://doi.org/10.1007/978-0-387-34878-0_18

[35] M. S. Mahmoud, M. M. Hamdan, Fundamental issues in networked control systems, IEEE/CAA Journal of Automatica Sinica 5 (5) (2018) 902–922. doi:10.1109/JAS.2018.7511162.

[36] R. G. Sanfelice, Networked hybrid dynamical systems: Models, specifications, and tools, 2018.

[37] P. Tallapragada, N. Chopra, Event-triggered dynamic output feedback control of lti systems over sensor-controller-actuator networks, in: 52nd IEEE Conference on Decision and Control, 2013, pp. 4625–4630. doi:10.1109/CDC.2013.6760613.

[38] T. Sanislav, S. Zeadally, G. Mois, H. Fouchal, Multi-agent Architecture for Reliable Cyber-Physical Systems (CPS) (Pediswesa) (2017) 2–7.

[39] Q. Zhu, L. Bushnell, T. Başar, Resilient Distributed Control of Multi-agent Cyber-Physical Systems, Springer International Publishing, Heidelberg, 2013, pp. 301–316. doi:10.1007/978-3-319-01159-2_16. URL https://doi.org/10.1007/978-3-319-01159-2_16

[40] D. Calvaresi, M. Marinoni, A. Sturm, M. Schumacher, G. Buttazzo, The challenge of real-time multi-agent systems for enabling iot and cps, 2017, pp. 356–364. doi:10.1145/3106426.3106518.

[41] A. Ali, Towards the design of cyber-physical system via multi-agent system technology, International Journal of Scientific and Engineering Research 7 (10 2016).

[42] E. Sultanovs, A. Romanovs, Centralized healthcare cyber-physical system's data analysis module development, 2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering, AIEEE 2016 - Proceedings (November 2016) (2017) 1–4. doi:10.1109/AIEEE.2016.7821826.

[43] M. Mitici, J. Goseling, M. Graaf, R. Boucherie, Decentralized vs. centralized scheduling in wireless sensor networks for data fusion, 2014, pp. 5070–5074. doi:10.1109/ICASSP.2014.6854568.

[44] D. Sauter, T. Boukhobza, F. Hamelin, Decentralized and autonomous design for fdi/ftc of networked control systems, in: H.-Y. Zhang (Ed.), Fault Detection, Supervision and Safety of Technical Processes 2006, Elsevier Science Ltd, Oxford, 2007, pp. 138–143. doi:https://doi.org/10.1016/B978-008044485-7/50024-5.
URL https://www.sciencedirect.com/science/article/pii/B9780080444857500245

[45] S. A. Alavi, X. Li, K. Mehran, Chapter two - delay resilient networked control with application to microgrids: Multi-agent and fuzzy control approaches, in: M.-H. Khooban, T. Dragicevic (Eds.), Control Strategy for Time-Delay Systems, Emerging Methodologies and Applications in Modelling, Academic Press, 2021, pp. 35–68. doi:https://doi.org/10.1016/B978-0-32-385347-7.00007-9.
URL https://www.sciencedirect.com/science/article/pii/B9780323853477000079

[46] A. Rabinovich, STOCHASTIC TIMED AUTOMATA, Logical Methods in Computer Science 10 (1) (2014) 1–73. doi:10.2168/LMCS-10.

[47] A. David, P. G. Jensen, K. G. Larsen, M. Mikučionis, J. H. Taankvist, Uppaal stratego, in: C. Baier, C. Tinelli (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 206–211.

[48] C. Zhu, M. Chai, Y. Lu, Y. Guo, Service oriented architecture design of energy consumption information system about petroleum enterprise, in: 2013 International Conference on Computational and Information Sciences, 2013, pp. 1174–1177. doi:10.1109/ICCIS.2013.311.

[49] M. Richards, Layered Architecture, 2009. doi:10.1007/978-0-387-39940-9_2941.

[50] IBM Cloud Education, What is SOA (Service-Oriented Architecture)? — IBM (2021).
URL https://www.ibm.com/cloud/learn/soa

[51] M. M. Alani, Guide to OSI and TCP/IP Models, Springer Publishing Company, Incorporated, 2014.

[52] P. Johnson, An OSI Model for Cloud - Cisco Blogs (2017).
URL https://blogs.cisco.com/cloud/an-osi-model-for-cloud

Kanwal Ashraf is a PhD student at Tallinn University of Technology, Estonia. Kanwal received her Joint-International Master's degree in Smart System Integration from Heriot-Watt University, UK, USN, Norway and BME, Hungary in 2020. She holds a bachelor's degree in electrical engineering from the University of the Punjab, Pakistan. Her research topics include the application of Wireless communication in CPSs and Co-Design of Wireless Networked Control Systems for bioanalytical applications.

Yannick Le Moullec received the M.Sc. degree in electrical engineering from Université de Rennes I, Rennes, France, in 1999 and the PhD degree in Electrical Engineering from Université de Bretagne Sud, Lorient, France, in 2003. From 2003 to 2013, he successively held Postdoc, Assistant, and Associate Professor positions with Aalborg University, Aalborg, Denmark. He then joined Tallinn University of Technology, Tallinn, Estonia as a Senior Researcher and then on a professorship of Cognitive Electronics. He has supervised eleven Ph.D. theses and nearly 60 M.Sc. theses. He was Co-PI for the H2020 COEL ERA-Chair project. His research interests include HW/SW codesign, embedded systems, reconfigurable systems, and IoT.

Tamas Pardy received the M.Sc. degree in info-bionics engineering from Peter Pazmany Catholic University, Budapest, Hungary, in 2014, and the Ph.D. degree in electronics and telecommunication from Tallinn University of Technology, Tallinn, Estonia, in 2018. He is currently a senior researcher at Tallinn University of Technology, focusing on flow- and temperature-control of Lab-on-a-Chip devices.

Toomas Rang was born on November 10, 1951, in Tallinn, Estonia. He graduated from Tallinn Technical University (TTU), Estonia as an Electronics Engineer in 1975. In 1981 he successfully defended the PhD degree in Electronics at the Hungarian Academy of Sciences. From 1984/1985, he held a Postdoc position at Saarland University, Germany. In 1996 he was elected for the professor position in Electronics Design at TTU. In 1998 he was elected to the head of the Department of Electronics. Since 2020 he is the Professor Emeritus at TTU. He is a senior member of IEEE. He has successfully supervised fifteen Ph.D. theses and is author of about 120 scientific publications. His research interests cover the topics from wide bandgap-based semiconductor devices and metallization technologies, till the microfluidics-based Lab-on-Chip devices.