Automated Dimensioning of Networked Labs-on-Chip

Andreas Grimmer *Student Member, IEEE,*
Werner Haselmayr *Member, IEEE,* and Robert Wille *Senior Member, IEEE*

***Abstract*—*Two-phase flow microfluidics* is a sophisticated and frequently applied *Labs-on-Chip* technology as they allow to automatically conduct medical/biochemical experiments. In this technology, small volumes of reagents, so-called *droplets*, flow in an immiscible continuous flow inside closed channels making it particularly biocompatible. In the recent past, this technology was extended by a concept allowing to *passively navigate* droplets through the system – leading to so-called *Networked Labs-on-Chips* (NLoCs). After the design of an NLoC architecture which defines the comprising connectivity between components and, by this, how the considered medical/biochemical experiments are supposed to be realized, the question remains how to properly dimension the used components, i.e. especially how to dimension the used channels. However, this is a challenging task which is conducted manually thus far and frequently leads to specifications that do not work as intended. In this work, we are addressing this issue by providing the designer with methods that allow to (1) automatically validate whether a chosen specification of an NLoC indeed works as intended as well as (2) automatically dimension NLoCs. Case studies demonstrate the importance and usefulness of the proposed methods for determining proper specifications of NLoCs.**

***Index Terms*—Networked Labs-on-Chips, passive droplet routing, droplet microfluidics, two-phase flow microfluidics, dimensioning, specification.**

I. Introduction

Labs-on-Chips (LoCs) enable the miniaturization, integration, automation, and parallelization of complex medical/biochemical experiments [1]. A sophisticated technology is based on *two-phase flow microfluidics* where samples e.g. of DNA, proteins, cells, as well as organisms [2] are injected into so-called *payload droplets* (i.e. in volumes in the order of few micro- to pico-liters). Then, these droplets flow in an immiscible continuous flow inside closed channels and are passed through modules executing operations such as mixing, heating, and detecting – realizing the desired medical/biochemical experiment. Furthermore, the use of droplets allows for a long-term incubation and storage without evaporation [1]. Finally, the extension to *passively navigate* these droplets through the system allows for different complex experiments in the same chip [3] as well as for screening of different antibiotics and concentrations [4]. This eventually results in a concept known as *Networked Labs-on-Chip* (NLoC, [5]) (also called *Hydrodynamic Controlled Microfluidic Network* [3]).

Each design of an NLoC starts with a set of medical/biochemical experiments. The first step is to determine an architecture by defining the required set of operations and by connecting them so that all desired experiments can be realized. This results in an architecture with multiple paths through which the droplets can be navigated and, by this, the different experiments can be realized. After this, the specification of the components used in the architecture (i.e. the modules, channels, and pumps) has to be determined – i.e. the NLoC needs to be *dimensioned*.

The specifications of all components and their connections determine the flow of droplets. Especially, the specification of channels (i.e. their resistances) can be varied in a broad bandwidth and, by this, their dimensioning constitutes a significant challenge. In fact, improper specifications can cause

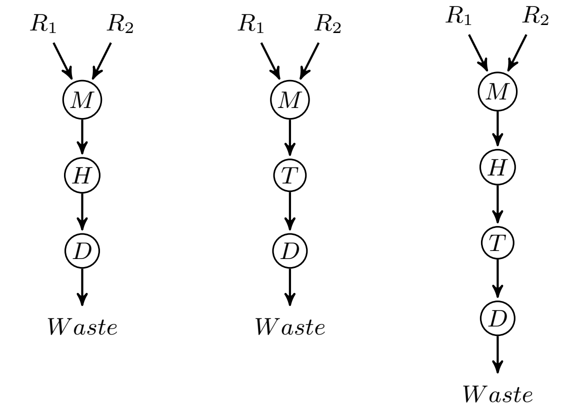
- the flow in channels/modules to be in the wrong direction or
- the time a droplet requires to pass a channel/module to be too long/short.

However, for dimensioning a channel, a huge number of constraints and dependencies have to be considered and already slightly changing e.g. the resistance of a single channel may change the behavior of the entire NLoC system. Additionally, no dedicated tool support exists yet for completing this specification. Hence, designers are usually left alone during this crucial step, which frequently yields specifications that do not work as intended.

In this work, we are addressing these problems by introducing first automatic methods that aid designers in the specification of NLoCs[1] – especially in the dimensioning of channels. More precisely, we propose methods which automatically allow to (1) validate whether a manually derived specification indeed works as intended, i.e. fulfills certain objectives as well as (2) conduct the dimensioning to obtain a proper specification. Both methods are based on an established and empirically validated physical model [6], which allows to describe the flow in NLoCs. This model utilizes an analogy between microfluidics and electric circuits and is commonly used for simulating droplet-based microfluidic networks [6]–[14].

Case studies show that both methods significantly aid the designer in the process of determining a precise specification for NLoCs. To this end, we provide a designer having expert knowledge with these two methods. They enabled him/her to quickly check and refine initial specifications as well as to efficiently determine a specification which works as intended. After a valid specification is obtained, the next step in the design flow is to derive a physical design.

[1] Note that the proposed methods are not limited to NLoCs, but are applicable for other microfluidic circuits and systems as well.



(a) Experiment 1 (b) Experiment 2 (c) Experiment 3

Fig. 1: Experiments

The remainder of this paper is structured as follows: The next section reviews how experiments are realized on an NLoC architecture as well as how to route droplets along paths through the architecture. Afterwards in Section III, the general approach for dimensioning an NLoC and the resulting challenges are pointed out. In Section IV and V, the two automated methods for validating specifications and automatically determining a proper specification are introduced. They are afterwards discussed in Section VI before results obtained using both methods are summarized in Section VII. Finally, the paper is concluded in Section VIII.

II. NLoC-Background: From Experiments to their Execution

In this section, we review the current state-of-the-art how to get from initially given medical/biochemical experiments to an NLoC architecture and, finally, how to conceptually route droplets so that an experiment is executed. The architecture defines the necessary components and their comprising connectivity. Therefore, it provides the basis on which the dimensioning of all components considered in this work is conducted – eventually yielding a full specification of the microfluidic network.

A. Experiments

The starting point of each NLoC design is a set of medical/biochemical experiments to be realized. An experiment is a sequence of operations which have to be executed on samples. These samples are provided in terms of discrete droplets and are called *payload droplets*. Possible operations can e.g. be mixing, splitting, delaying, incubating, detecting, or heating [15]–[19].

Example 1. *Let's assume that three different experiments have to be realized which are provided in terms of sequencing graphs as shown in Figs. 1a–1c. In the experiment shown in Fig. 1a, two reagents R_1 and R_2 are fused into a single payload. Afterwards, a mixing operation (M) followed by a heating operation (H) and a detecting operation (D) is to be executed on the resulting payload. The experiment shown in Fig. 1b is similar to the one from Fig. 1a, except that, instead of the heating, a delaying operation (T) is to be executed. Finally, in the experiment shown in Fig. 1c, the mixed payload has to be heated and delayed before the detecting operation shall be conducted.*

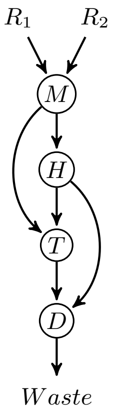


Fig. 2: Architecture

B. Determining the Architecture

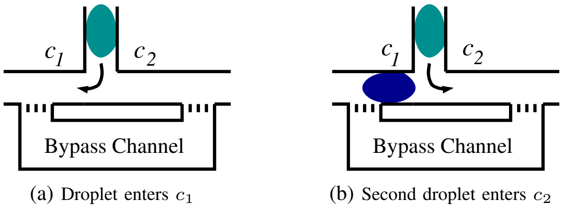
Having the experiments to be realized, the first design step covers the determination of an architecture. Therefore, an automatic method was presented in [20]. This method determines the required operations and their order for supporting the execution of the desired experiments. In order to allow for a cost-effective architecture, operations can be re-used for different experiments. Formally, an architecture is a directed, acyclic graph with all edges directed from the input reagents forming the payloads to the outlets of the NLoC (e.g. the waste chambers). A node of the architecture represents a module executing an operation. An edge represents a connection between nodes which is implemented as a microfluidic channel.

Example 2. *In order to execute the experiments shown in Figs. 1a–1c, a single mixing-, heating-, delaying-, and detecting-operation is sufficient. The automatic method of [20] connects these operations so that sequences realizing all three considered experiments result – yielding an architecture as described by the graph shown in Fig. 2.*

A node of the architecture can have one, two, or more outgoing edges. The reachable nodes of those edges represent possible modules which are executed next on the payload. Here, a mechanism to decide which module is executed next is needed. To this end, so-called *bifurcations* are utilized [5], [21]. Based on that, it is left to define how payloads actually navigate through the architecture, which is discussed next.

C. Routing of Droplets

Given an architecture, payloads can take different paths and, by this, execute different experiments. The paths are realized by bifurcations, which split a channel into two successor channels [5], [21]. In the architecture, when a node has



(a) Droplet enters c_1 (b) Second droplet enters c_2

Fig. 3: Bifurcation

multiple outgoing edges, a bifurcation is used to split the channel into two successor channels. These successor channels have different *fluidic resistances* which are mainly defined by their *geometries* [22], e.g. the longer the channel the higher the resistance and the smaller the section the higher the resistance. When a *single* droplet arrives at a bifurcation, it flows into the channel with the lower resistance [8] (the so-called *default successor*). Hence, the successor channels of bifurcations are designed in a way that they have different hydraulic resistances – favoring one particular successor channel. In order to (partially) decouple the resistances of the successor channels with the rest of the microfluidic network, a bifurcation is equipped with a so-called *bypass channel* [23], which cannot be entered by droplets. Technically, a bypass has a very low fluidic resistance and, hence, levels out the pressure gradients between the two successors channels.

Example 3. *Fig. 3a shows a bifurcation. The channel c_1 is the default successor since its length is shorter than c_2. A* single *droplet therefore flows into the default successor channel.*

However, a droplet itself increases the resistance of a channel when flowing through it [5], [21]. This physical effect is used at bifurcations in order to route a payload along a path which is not solely composed of default successors. More precisely, whenever a non-default successor should be taken, a so-called *header droplet* (which does not contain any biological information and is only used to navigate payloads) is injected so that it arrives right before the payload at the bifurcation. Then, the header takes the default successor of the bifurcation and, by its flow, increases the fluidic resistance of that channel (it temporary "blocks" the default successor). Accordingly, the following payload does not take the default successor but enters the other channel (which now has a lower fluidic resistance). This routing concept and its design guidelines, addressing schemes, validation through simulations, and initial experimental tests have been presented in [3], [5], [21], [24]–[26].

Example 4. *Fig. 3b shows a bifurcation where a header droplet flows in the default successor channel c_1. As a consequence, the closely following payload will enter channel c_2.*

This physical effect allows to route payloads through different paths of the architecture and, eventually, allow for the realization of different experiments. Therefore, a dedicated sequence of header and payload droplets is injected which ensures that headers accordingly arrive at bifurcations and "block" the respective channels at the right time. Moreover, also the design of the architecture itself is crucial for the droplet sequence generation, which is addressed in [27] by employing the discrete model of [28].

III. Dimensioning Networked Labs-on-Chip

After reviewing the design of an NLoC architecture and how droplets can be routed along different paths, we now consider the question how to derive a specification of an NLoC. Here, the NLoC designer has to determine a specification for all components used in the architecture, i.e. the designer has to *dimension* the respective components. In the following, we describe how designers accomplish this dimensioning up to now and particularly discuss the challenges of this process. Afterwards, we review the flow distribution in NLoCs needed to understand these challenges.

A. Components and Challenges

For the specification, actual components realizing the given architecture have to be selected. This includes

- a *pump* for producing a continuous flow which drives the droplets through the system,
- *droplet generators* for injecting payloads and headers,
- *modules* for executing operations on the payloads,
- *bifurcations* for allowing the droplets to take different paths, as well as
- *channels* for connecting the components (pumps, modules, bifurcations, and outlets) with each other.

The pump, droplet generators, modules, and bifurcations are implicitly defined by the experiments to be executed and a variety of physical realizations exists for this purpose. More precisely:

The *pump* is a (usually external) device which produces a continuous flow through the NLoC.

The *droplet generation* is realized with *droplet-on-demand* components. Here, a second pump produces a force on the dispersed phase and the droplets are generated with internal valves [29], external valves [30], or with pressure pulses [31]–[33]. These components allow to injected droplets at dedicated times and, hence, allow to generate the required droplet sequence consisting of the payload and header droplets.

In order to execute operations, a rich collection of *modules* is available including e.g. implementations for fusing, mixing, sorting, delaying, incubating, and detecting of droplets [2], [15]–[19], [34], [35]. These modules are supposed to be applied to payloads only, while headers are forwarded[2].

Finally, in case an operation of the architecture has multiple successors, a *bifurcation* is applied to allow for droplets to take different paths as already reviewed in Section II-C.

Example 5. *Consider again the NLoC design represented by the architecture provided in Fig. 2. Choosing the respective physical components yields a* partial *specification of the NLoC as shown in Fig. 4. Here, a pump is used for producing the continuous phase, two droplet-on-demand generators are used*

[2] Forwarding headers is conducted using a droplet *sorter*, which can distinguish between payloads and headers by their different volumes [36].

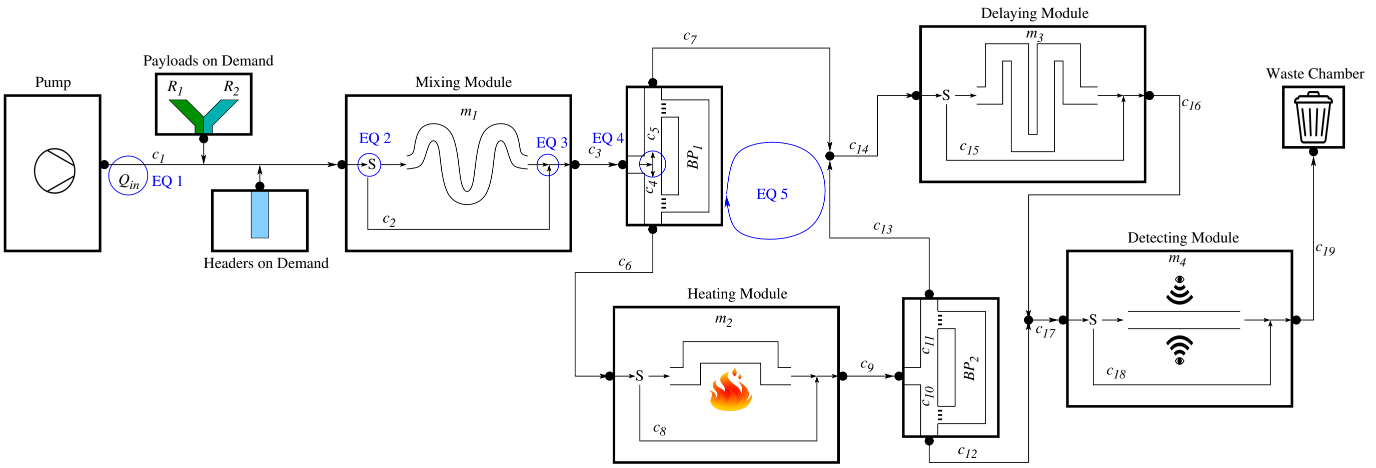


Fig. 4: Partial specification of the architecture from Fig. 2

to inject payloads and headers. The operations are realized by the corresponding modules (denoted by m_1, m_2, m_3, and m_4) including channels for forwarding headers (denoted by c_2, c_8, c_{15}, and c_{18}). Finally, two bifurcations (with successor channels denoted by c_4, c_5, c_{10}, and c_{11}) are applied in order to realize the different paths.

Having this partial specification, it is left to define the required connections, i.e. to properly dimension the channels. This however constitutes a challenge, since their resistances significantly affect the flow of the droplets. In the currently applied design flow, the channels are defined based on the designer's experience. While doing that, designers have to take a huge number of constraints and dependencies into consideration – already slightly changing e.g. the specification of a single channel may change the behavior of the entire NLoC system. As a consequence, designers often cannot grasp all effects and dependencies anymore. More precisely, because of improper specifications of channels, NLoC realizations may result in which

- droplets flow against the intended direction or
- droplets pass a channel/module too slowly (critical when e.g. a payload just passed a heating module and needs to be analyzed by a following detector module without cooling down) or too quickly (e.g. when the channel is used as delay line).

In order to explain why, when, and how these problems occur, we provide the physical basis describing the flow distribution in the following. Based on that, we afterwards propose methods which allow designers to (1) automatically validate whether their chosen specification indeed works as intended (i.e. avoid problems as discussed above) as well as (2) automatically conduct the dimensioning to obtain a proper specification.

B. Flow Distribution

To understand, why, when, and how problems as discussed above occur, it is essential to understand how droplets flow through an NLoC system. Therefore, we review the flow distribution in NLoCs using an established and empirically validated model based on an electrical duality (a comprehensive review is given in [6]) in the following. This electrical duality is commonly used for designing and modeling such devices, e.g. in [7], [37]–[39].

Briefly, the movement of the droplets depends on the applied pump, which produces a flow of the continuous phase, as well as the channels and modules which change the distribution of that flow depending on their resistances and their arrangement. More precisely, the dimensionless Reynolds number (Re) defines the ratio of inertial forces to viscous forces (i.e. $Re = \rho\, u\, L/\mu$, where ρ is the density, u is the linear velocity, L is a characteristic length, and μ is the viscosity). In microfluidics, this number is generally small ($Re \leq 1$) [6], [35] due to the small channel sections and relatively small flow rates. Hence, the inertial effects (i.e. the gravity, separation, secondary flow and turbulence) are negligible. This allows to describe the flow using the following parameters and relations:

- Each channel/module poses a *fluidic resistance* R_c/R_m (in $[Pa\, s/m^3]$) for the flow.
- The volume of the continuous fluid through the channel/module per time unit is called *volumetric flow rate* Q_c/Q_m (in $[m^3/s]$).
- The difference of the pressure between the input and the output of the channel/module is called *pressure gradient* ΔP_c/ΔP_m (in $[Pa]$).

The *Hagen-Poiseuille* equation [40] describes the relation between these parameters by $\Delta P = R\, Q$. This is similar to the well-known *Ohm's law* $V = R\, I$ from electronics, where the fluidic resistance, the volumetric flow, and the pressure gradient are counterparts of the resistance R of a resistor, the current I, and the voltage V, respectively. In fact, the interplay between these flow parameters can directly be represented by the Ohm's law and, hence, the same rules as in electrical circuits can also be employed for NLoCs [6].

Based on that, we can determine the fluidic flow in all channels/modules, by considering (1) what pump has been chosen in order to produce the force enabling the droplet flow as well as (2) how have the channels and modules been defined and arranged within the NLoC.

(1) Effect of the Pump: The pump produces the continuous flow of a fluid with viscosity μ (given as dynamic viscosity in $[Pa\,s]$). To this end, designers can choose between two different pump realizations: A *syringe pump* produces a constant volumetric flow rate (the electrical counterpart is a current source) and a *peristaltic pump* produces a pressure gradient (the electrical counterpart is a voltage source). This yields a particular volumetric flow rate Q_{in} or a particular pressure gradient ΔP_{in}, respectively, which is applied to the NLoC system.

(2) Effect of the Channels/Modules: The resulting flow applied by the pump is distributed through the NLoC system depending on the definition and arrangement of all its channels/modules. Each module and channel poses a resistance to the flow affecting its distribution. The resistances depend on the viscosity of the continuous fluid μ and the geometrical specification of the module/channel. For a module m, the resistance R_m is given in its specification. For a rectangular channel c, its resistance R_c can be determined by its section, i.e. its width w_c and height h_c, as well as its length l_c (all in $[m]$). More precisely [22], the resistance R_c of a rectangular channel c where the ratio h_c/w_c is less than 1, is accurately defined by

$$R_c = \frac{\alpha\,\mu\,l_c}{w_c\,h_c^3}, \tag{1}$$

where α denotes a dimensionless parameter defined as

$$\alpha = 12\left[1 - \frac{192\,h_c}{\pi^5\,w_c}\tanh\left(\frac{\pi\,w_c}{2\,h_c}\right)\right]^{-1}. \tag{2}$$

Furthermore, the resistance of channels slightly increases when it contains droplets [22]. This resistance increase allows the navigation of droplets at bifurcations as reviewed in Section II-C. Besides that, the overall NLoC system needs to be operated at a low Capillary number (Ca) describing the effect between viscous forces and surface tension ($Ca = \mu\,u/\gamma$, where μ is the viscosity, u is the speed of the flow, and γ is the surface tension). A low capillary number minimizes the droplets' surface areas and makes the droplets controllable, i.e. the system has to work in the *squeezing regime* which requires $Ca < 10^{-2}$ [41]–[43].

The information on the applied pump (providing either the input volumetric flow rate Q_{in} or the input pressure gradient ΔP_{in}) as well as on the definition and arrangement of channels/modules, allows to determine the fluidic flow in each channel/module by applying the *Kirchhoff's laws* [6], [44]. More precisely: Each channel/module has a fixed counting direction of the volumetric flow (usually the direction of the intended flow). Then, the two Kirchhoff's laws can be expressed for NLoCs as follows:

- *Kirchhoff's current law:* Each point in the NLoC realization where the volumetric flow splits into multiple components or merges into one component is called a *node*. The sum of flow rates into a *node* is equal to the sum of flows rates out of that node.
- *Kirchhoff's voltage law:* The sum of pressure gradients (expressed using $\Delta P = R\,Q$) around any closed *cycle* in the NLoC is zero. The direction of the volumetric flow is taken into account in the summands.

These rules allow to define an equation system which can be used to determine all flow rates Q_c/Q_m and, therefore, also all pressure gradients ΔP_c/ΔP_m of all channels and modules.

Example 6. *Consider again the partial NLoC specification as shown in Fig. 4. Additionally, assume that the counting direction of the volumetric flows for all channels and modules follow the intended flow direction from the pump to the waste chamber (for the channels, this is indicated by the arrows in Fig. 4). Then, the flow rates can be determined by the following equation system (note that, only an excerpt representing the part highlighted by five blue circles in Fig. 4 is shown):*

Eq 1: $Q_{in} = Q_{c_1}$
Eq 2: $Q_{c_1} = Q_{m_1} + Q_{c_2}$
Eq 3: $Q_{m_1} + Q_{c_2} = Q_{c_3}$
Eq 4: $Q_{c_3} = Q_{c_4} + Q_{c_5}$
...
Eq 5: $Q_{c_7}\,R_{c_7} - Q_{c_{13}}\,R_{c_{13}} - Q_{c_{11}}\,R_{c_{11}} - Q_{c_9}\,R_{c_9}$
$\quad - Q_{m_2}\,R_{m_2} - Q_{c_6}\,R_{c_6} + Q_{BP_1}\,R_{BP_1} = 0$
...

The resulting flow rates can eventually be used to explain why, when, and how droplets move in the opposite direction or are too slow/fast. This is covered in the next two sections.

IV. Validating the Specification

Using the descriptions of the physical behavior provided above as basis, a method is proposed which automatically checks whether a given specification of an NLoC shows any of the problems discussed above. To describe the method, recall that, as discussed in Section III-A, the used pump, droplet generators, modules, and bifurcations are predefined by the experiment and the designer is mainly confronted with the task of properly dimensioning the channels – especially their resistances.

Example 7. *Consider again the NLoC architecture to be realized as shown in Fig. 2 as well as the resulting partial specification as shown in Fig. 4. Let's additionally assume that the designer uses a syringe pump producing a constant volumetric flow rate $Q_{in} = 50 \cdot 10^{-12}\,m^3/s$ of water having a viscosity of $\mu = 10^{-3}\,Pa\,s$, a density of $\rho = 1\,g/ml$ and a surface tension of $\gamma = 72.75\,mN/m$ (at approx. 20°C). This viscosity in combination with the specification of the selected modules and bifurcations allows to determine the resistances of the following entities (given in $10^{12} Pa\,s/m^3$)[3]:*

$R_{m_1}, R_{m_2}, R_{m_3}, R_{m_4}$	$R_{c_2}, R_{c_8}, R_{c_{15}}, R_{c_{18}}$	$R_{c_4}, R_{c_{10}}$	$R_{c_5}, R_{c_{11}}$
5.2	5.4	0.8	0.9

Now, let's assume that the designer (e.g. based on his/her experience or on a purely trial-and-error basis) dimensions all remaining channels with the same dimensions, i.e. a width of $w = 50\,\mu m$, a height of $h = 50\,\mu m$, and a length of

[3] Note that these resistances are chosen in a way so that they are uniform and suited to discuss the considered problems.

$l = 200\,\mu m$. *This allows to determine the resistance of these channels using Eq. 1 (given in* $10^{12}\,Pa\,s/m^3$*):*

$R_{c_1}, R_{c_3}, R_{c_6}, R_{c_7}, R_{c_9}, R_{c_{12}}, R_{c_{13}}, R_{c_{14}}, R_{c_{16}}, R_{c_{17}}, R_{c_{19}}$
0.9

Conducting the steps illustrated in the example would usually complete the specification. However, as discussed above, the chosen dimensions of the channels may yield a specification that does not work as intended. Thus far, designers had the choice to simulate their mapping and to manually inspect if any of the problems discussed above occurs. But here, we provide a method which is capable of automatically validating the specification.

First let's consider the problem where droplets flow in the opposite direction:

- *Objective 1 – A droplet flows in the opposite direction:* This is the case when at least one channel/module exists in which its determined flow rate Q_c/Q_m is negative.

Example 8. *Consider again the partial NLoC specification as shown in Fig. 4. Solving the equation system as derived in Example 6 together with the choices of the designer as specified in Example 7, yields, among others, the following flow rates (given in* $10^{-12}\,m^3/s$*):*

Q_{c_1}	Q_{m_1}	Q_{c_2}	Q_{c_3}	Q_{c_4}	Q_{c_9}	$Q_{c_{10}}$	$Q_{c_{12}}$	$Q_{c_{13}}$	$Q_{c_{17}}$	Q_{m_2}	Q_{BP_1}
50	31	19	50	23	13	6.8	36.2	-23.5	50	7.8	-10

Since the flow rate in channel c_{13} *is negative, a violation of Objective 1 is observed for this channel*[4]*. This clearly shows that the choices by the designer yield an NLoC specification which does not work as intended.*

Besides that, using a similar scheme, let's consider the problem where droplets take a too long/short time:

- *Objective 2 – A droplet passes a channel too slowly/quickly:* This is the case when at least one channel/module exists in which its determined flow rate Q_c/Q_m yields a speed resulting in a duration t_c/t_m which is larger/smaller than a time limit $T_\triangle$.

In order to determine the durations, again, the flow rates can be used. In fact, by dividing the flow rate by the section of the corresponding channel/module, the speed (in m/s) of the flow can be determined, i.e. by

$$u_c = \frac{Q_c}{w_c\,h_c} \quad \text{or} \quad u_m = \frac{Q_m}{w_m\,h_m}. \tag{3}$$

Then, these speeds can be used to approximate[5] the duration (in s) a droplet requires in order to pass a channel/module, i.e. by

$$t_c = \frac{l_c}{u_c} \quad \text{or} \quad t_m = \frac{l_m}{u_m}. \tag{4}$$

Example 9. *Let's assume that droplets should move from the heating module to the detecting module in less than* $T_\triangle = 1\,s$ *(e.g. to prevent the droplet from cooling down before it gets analyzed). As this requires the droplet to flow through channels* c_9*,* c_{10}*,* c_{12}*,* c_{17}*, the sum of the corresponding durations must be less than that. Since according to Example 8,* $Q_{c_9} = 13$*,* $Q_{c_{10}} = 6.8$*,* $Q_{c_{12}} = 36.2$*, and* $Q_{c_{17}} = 50$ *(given in* $10^{-12}\,m^3/s$*), the respective durations are* $t_{c_9} = 39ms$*,* $t_{c_{10}} = 65ms$*,* $t_{c_{12}} = 14ms$*, and* $t_{c_{17}} = 10ms$*. This sums up to* $128\,ms$ *and, hence, validates that the choice by the designer yields an NLoC specification fulfilling at least this objective.*

Finally, it has to be checked whether the Reynlods number and the Capillary number fall into the desired ranges (cf. Section III-B).

Example 10. *The maximal speed of the continuous phase is determined by dividing the maximal flow rate (cf.* $50 \cdot 10^{-12}\,m^3/s$ *in Example 8) by the cross section of the respective channel (*$50\,\mu m \times 50\,\mu m$*) and is equal to* $0.02\,m/s$*. In this example, the Reynolds number is equal to* $Re = 1$ *and the Capillary number is equal to* $Ca = 2.75 \cdot 10^{-4}$ *(the characteristic length L for a squared channel is its width [6]), i.e. both are within the desired ranges.*

Overall, the proposed method is capable of automatically validating choices by the designer.

V. Automatic Dimensioning

An automatic validation of a specification clearly supports the designer. Nevertheless, it does not relieve the designer from the burden to make choices until a proper specification has been found. Moreover, it may even be possible that, using the existing set of modules as well as the current arrangements of channels/modules, no proper specification is possible, i.e. independent of the choices of the designer, one of the objectives mentioned above might always fail. For example, this can easily happen if the flow rate/pressure gradient produced by the pump is too low so that the timing objectives cannot be ensured.

Using the physical basis introduced in Section III-B also allows to aid the designer in these issues. In fact, by leaving the channels' resistances free, still a result can be obtained from the equation system. As this however may again include values violating one of the objectives from above, the equation system has to be extended by further equations.

First, we have to ensure reasonable resistances for all unspecified channels c. Therefore, equations are added enforcing a minimum R_{min} and a maximum resistance R_{max}, i.e.

$$R_{min} \leq R_c \leq R_{max}. \tag{5}$$

Example 11. *Consider again the partial NLoC specification as shown in Fig. 4. For all unspecified connections, the method enforces a minimum and maximum resistance using the inequality* $0.9 \leq R_c \leq 15.4$ *(given in* $10^{12} Pa\,s/m^3$*).*

Then, the objectives for obtaining a proper NLoC specification are added: For Objective 1, equations for all channels

[4] Note that the flow rates in bypass channels are irrelevant as they cannot be entered by droplets.

[5] Note that this is an approximation, since droplets increase the resistance in channels/modules. Exact durations can be afterwards obtained by simulating the injected droplet sequence.

and modules enforcing the intended flow direction are added, i.e.

$$Q_c \geq 0 \text{ and } Q_m \geq 0. \tag{6}$$

Note that these constraints are not added for bypass channels of bifurcations because droplets cannot enter these channels.

For Objective 2, an equation enforcing that droplets will always pass a channel/module below/above a given timing threshold (namely T_Δ) is added. To this end, the equation for determining the time a droplet takes to pass/execute a channel/module is employed.

For calculating the speed and the time a droplet requires to pass a channel, the section of the channel (i.e. its width and height) is needed. This section is usually fixed over the whole chip in order to minimize the (production) complexity (see e.g. [39]). This allows to express the length l_c of a channel using its resistance. By inserting this equation transformation into Eq. 3 and Eq. 4, the required time can be restricted by

$$\frac{R_c\, w_c^2\, h_c^4}{\alpha\, \mu\, Q_c} \leq T_\Delta \quad \text{or} \quad \frac{R_c\, w_c^2\, h_c^4}{\alpha\, \mu\, Q_c} \geq T_\Delta. \tag{7}$$

For modules, its geometry specification and the flow rate allows to determine the required time, i.e. by

$$\frac{l_m\, w_m\, h_m}{Q_m} \leq T_\Delta \quad \text{or} \quad \frac{l_m\, w_m\, h_m}{Q_m} \geq T_\Delta \tag{8}$$

Accordingly, if we need an upper/lower bound for a sequence of channels/modules, the respective times of the single components can be summed up and, afterwards, restricted.

Example 12. *Let's assume the same objectives as specified in Example 8 and Example 9 are applied. In order to ensure these, the method adds for all channels and modules the inequalities*

$$Q_c \geq 0\, m^3/s \;\; \textit{and} \;\; Q_m \geq 0\, m^3/s. \tag{9}$$

The upper time limit $T_\Delta = 1\, s$ is ensured by

$$\sum_{c \in \{c_9, c_{10}, c_{12}, c_{17}\}} \frac{R_c\, w_c^2\, h_c^4}{\alpha\, \mu\, Q_c} \leq 1\, s. \tag{10}$$

Finally, the ranges of the Reynlods number and the Capillary number have to be restricted (cf. Section III-B). Therefore, the method adds for all channels and modules the inequalities

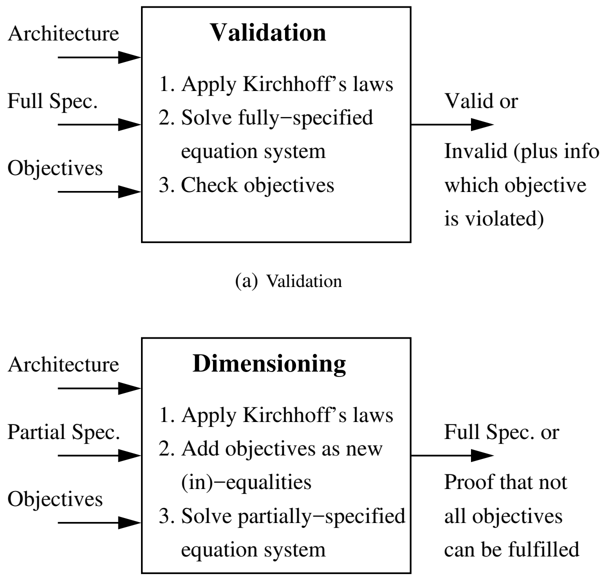
$$\frac{\rho\, Q_c\, L}{\mu\, w_c\, h_c} \leq 1 \quad \text{and} \quad \frac{\rho\, Q_m\, L}{\mu\, w_m\, h_m} \leq 1 \tag{11}$$

for the Reynolds number as well as

$$\frac{\mu\, Q_c}{\gamma\, w_c\, h_c} < 10^{-2} \quad \text{and} \quad \frac{\mu\, Q_m}{\gamma\, w_m\, h_m} < 10^{-2} \tag{12}$$

for the Capillary number.

Successfully solving the resulting equation system yields values for all variables – including values for the resistances and flow rates of each channel. From this, the respective dimensions of the channels can be derived and the specification is completed. In contrast, if it can be proven that no solution for this equation system exists, it has been shown that the partial specification does not allow to fulfill all objectives.


Architecture
Full Spec.
Objectives
Validation
1. Apply Kirchhoff's laws
2. Solve fully−specified equation system
3. Check objectives
Valid or
Invalid (plus info which objective is violated)
Architecture
Partial Spec.
Objectives
Dimensioning
1. Apply Kirchhoff's laws
2. Add objectives as new (in)−equalities
3. Solve partially−specified equation system
Full Spec. or
Proof that not all objectives can be fulfilled

(a) Validation

(b) Automated Dimensioning

Fig. 5: Overview of the proposed methods

Example 13. *Recall that the designer originally dimensioned all remaining channels with the same resistance of $0.9 \cdot 10^{12}\, Pa\, s/m^3$ (cf. Example 7). Solving the equation system resulting from the steps above yields slightly different resistances, namely (given in $10^{12} Pa\, s/m^3$):*

R_{c_1}	R_{c_3}	R_{c_6}	R_{c_7}	R_{c_9}	$R_{c_{12}}$	$R_{c_{13}}$	$R_{c_{14}}$	$R_{c_{16}}$	$R_{c_{17}}$	$R_{c_{19}}$
0.9	0.9	0.9	13.9	0.9	4.4	0.9	0.9	0.9	0.9	0.9

In contrast to the original specification by the designer, these resistances fulfill the first objective, i.e. no droplet flows in the opposite direction. In fact, the assigned resistances lead to a flow rate $Q_{c_{13}} = 12.5$ (in $10^{-12}\, m^3/s$) which is now positive. At the same time, this also ensures the timing objective from the heating to the detecting module. Overall, this yields a complete specification of the NLoC which is the basis for the physical design (i.e. the layout) of the chip.

VI. Application and Integration

In this section, we briefly discuss how the two methods proposed above can be applied in practice and how the resulting methods fit into the existing design flow of NLoCs.

A. Application

The inputs, conducted steps, as well as the output for both methods are summarized in Fig. 5. Their inputs are similar: Both take an architecture as input, which is then used to automatically determine an equation system using the Kirchhoff's laws. For the validation method (cf. Fig. 5a), the NLoC designer additionally provides a *full* specification of e.g. the pump, modules, *and* channels. For the automatic dimensioning method (cf. Fig. 5b), the NLoC designer only provides a *partial* specification, i.e. the resistances of channels are left unspecified. Finally, the NLoC designer formulates the objectives which have to be fulfilled.

The validation method solves a fully-specified equation system and afterwards checks all objectives. On the other hand, the automatic dimensioning method additionally adds the objectives in form of new (in)-equations and, then, determines a possible solution of an underdetermined system of equations (cf. the unspecified channels are free variables). If here an assignment to all variables is determined, it represents one possible specification.

Additionally, application-specific optimization criteria can be implemented by formulating optimization functions. Furthermore, both methods are open for new, application-specific objectives, e.g. for checking *Young-Laplace pressures*.

B. Integration into the Design Flow

The proposed methods perfectly advance the steps of the NLoC design between the initial determination of an architecture (cf. [20] and Section II-B) and the physical realization of an NLoC. While, thus far, the dimensioning has to be conducted manually (without any means to check whether the chosen values are somehow valid), automatic methods for validation and even generation are available now.

After completing the dimensioning of all components, the next step in the design flow of NLoCs (or generally for all microfluidic devices) is the *physical design*. In this step, the actual layout, i.e. the placement of all modules and the routing of the channels with the desired resistances, is created. The output of the physical design is usually a vector graphic, which can then be used for production (i.e. for a mask production using a soft-lithography process, for 3D-printing, or for laser-engraving).

However, during this process it may happen that the validated or generated resistances cannot be realized in an actual physical design (e.g. because the specified resistance of a channel results in a channel length which is too short to be properly routed). In this case, corresponding adjustments are directly conducted on the physical design. Then, the specification is re-validated using the methods proposed above. If again changes in the specification are necessary to fulfill all objectives, another iteration has to be conducted. This process of switching back to the specification and forth to the physical design is repeated as long as all objectives are fulfilled.

Note that the methods proposed in this work significantly help here as e.g. necessary changes can directly be conducted and validated immediately until the desired physical design results.

VII. Case Studies and Obtained Results

In order to demonstrate how the proposed methods improve the dimensioning of NLoCs, we conducted several case studies whose obtained results are summarized in this section. In the conducted case studies, we considered the dimensioning of five architectures which have been produced using the method proposed in [20]. The architectures are composed of 8 to 17 operations and 34 to 118 channels (entries in Table I provide detailed values). For these architectures, specifications were supposed to be determined.

TABLE I: Case Studies

	Setting	#Objective 1	#Objective 2	Time $[ms]$
Architecture with 8 modules and 34 channels				
Val.	Random	5	3	48
	Equal	0	0	47
	Designer (1 min)	0	0	1 × 51
Dim.	Automatic	0	0	68
Architecture with 10 modules and 67 channels				
Val.	Random	4	1	48
	Equal	4	0	49
	Designer (14 min)	0	0	16 × 69
Dim.	Automatic	0	0	105
Architecture with 12 modules and 82 channels				
Val.	Random	9	5	59
	Equal	12	1	60
	Designer (30 min)	2	0	19 × 60
Dim.	Automatic	0	0	164
Architecture with 15 modules and 101 channels				
Val.	Random	15	7	61
	Equal	15	1	60
	Designer (30 min)	1	0	23 × 75
Dim.	Automatic	0	0	734
Architecture with 17 modules and 118 channels				
Val.	Random	7	1	70
	Equal	17	2	70
	Designer (30 min)	5	1	22 × 77
Dim.	Automatic	0	0	330

Recall that, the specification of the pump, droplet generators, modules, and bifurcations is defined by the experiments to be executed – the actual challenges come with the specification of the channels. For a comprehensive evaluation, we considered four possible scenarios how these challenges are tackled:

- *Random*: All channels are dimensioned in a random fashion using values within reasonable intervals – hoping to determine a working solution by chance.
- *Equal*: All channels are dimensioned in the same (equal) fashion[6].
- *Explicit Design*: All channels are dimensioned by giving a designer with expert knowledge at most 30 minutes per architecture to derive a proper specification in a trial-and-error fashion (having the opportunity to constantly check the results using the method of Section IV).
- *Automatic Dimensioning*: All channels are dimensioned by the automatic method proposed in Section V.

In order to evaluate these scenarios, we implemented the two methods proposed above using Java. Afterwards, these methods have been used to validate and to automatically dimension the specification with respect to the two objectives, namely whether droplets flow in the opposite direction (this is considered the case when the flow rate Q_c/Q_m becomes negative in any channel/module) and whether droplets are too slow (in this evaluation, we required a droplet to pass all channels and modules in at most $T_\Delta = 1\,s$). Furthermore, the dimensionless Reynolds number and Capillary numbers are checked whether

[6] This is similar to the strategy illustrated in Example 7, where the channels are dimensioned with the same resistances.

they fall into the desired ranges (cf. Section III-B). These validations have been conducted on a 3.8 GHz Intel Core i7 machine with 32GB of memory running 64-bit Ubuntu 16.04.

Table I summarizes the obtained results. In the first three scenarios, the validation method described in Section IV (denoted "Val." in Table I) and in the last scenario the automatic dimensioning method described in Section V (denoted "Dim." in Table I) are applied. For each setting (i.e. NLoC design to be realized as well as scenario), the number of violations of the two objectives are listed. Besides that, we provide the total runtime needed to conduct a validation/automatic dimensioning. For the third scenario, we additionally provide the time spent by the designer (denoted in the column "Setting") as well as the total number of times the validation method has been applied (denoted by the multiplier in the column "Time").

As can be seen, dimensioning channel sizes is indeed a challenging task. Relying on random decisions, always improper specifications result which violate at least one objective. Also using equal channel resistances yields improper specifications for all NLoC designs except for the simplest one. Exploiting the expert knowledge performs better here. Although, the designer has to take a huge number of constraints and dependencies into consideration, the validation method significantly helps to quickly validate the choices. Overall, the designer managed to derive a proper specification for two architectures. However, for the three larger designs, the huge amount of constraints and dependencies made it impossible to manually derive a proper specification within 30 minutes.

The fourth considered scenario eventually solves this problem by automatically determining specifications fulfilling the required objectives. The results demonstrate that this method is capable of deriving a proper specification for all architectures within negligible runtimes.

Moreover, results obtained by the proposed methods have been tested using a simulator based on the model presented in Section III-B. These simulations confirmed that the full specifications (validated or obtained by the proposed methods) indeed allow to execute the considered experiments.

VIII. Conclusion

In this work, we considered how to get from a *Networked Labs-on-Chips* (NLoCs) architecture to a full specification of all components. We demonstrated that this is a challenging task as an improper dimensioning of channels will likely lead to NLoC specifications which are not working as intended. In order to support the NLoC designer in this task, we proposed methods for automatically (1) validating whether a given NLoC specification works as intended and (2) conducting the dimensioning to obtain a proper specification. Case studies confirmed that the proposed methods significantly aid the designer in the process: While a random or simple dimensioning hardly yielded an NLoC specification which works as intended, the validation method enabled the designer at least to quickly check and refine initial choices. The automatic dimensioning method eventually enabled him/her to efficiently obtain the desired specification in a push-button fashion.

References

[1] D. Mark, S. Haeberle, G. Roth, F. von Stetten, and R. Zengerle, "Microfluidic Lab-on-a-Chip platforms: requirements, characteristics and applications," *Journal of Chemical Society Reviews*, vol. 39, no. 3, pp. 1153–1182, 2010.

[2] B. Kintses, L. D. van Vliet, S. R. Devenish, and F. Hollfelder, "Microfluidic droplets: new integrated workflows for biological experiments," *Journal on Current Opinion in Chemical Biology*, vol. 14, no. 5, pp. 548–555, 2010.

[3] E. De Leo, L. Donvito, L. Galluccio, A. Lombardo, G. Morabito, and L. M. Zanoli, "Communications and switching in microfluidic systems: Pure hydrodynamic control for networking Labs-on-a-Chip," *Trans. on Communications*, vol. 61, no. 11, pp. 4663–4677, 2013.

[4] W. Haselmayr, M. Hamidović, A. Grimmer, and R. Wille, "Fast and flexible drug screening using a pure hydrodynamic droplet control," in *European Conference on Microfluidics*, 2018.

[5] E. De Leo, L. Galluccio, A. Lombardo, and G. Morabito, "Networked labs-on-a-chip (NLoC): Introducing networking technologies in microfluidic systems," *Journal of Nano Communication Networks*, vol. 3, no. 4, pp. 217–228, 2012.

[6] K. W. Oh, K. Lee, B. Ahn, and E. P. Furlani, "Design of pressure-driven microfluidic networks using electric circuit analogy," *Journal on Lab on a Chip*, vol. 12, no. 3, pp. 515–545, 2012.

[7] M. Schindler and A. Ajdari, "Droplet traffic in microfluidic networks: A simple model for understanding and designing," *Physical Review Letters*, vol. 100, no. 4, p. 044501, 2008.

[8] T. Glawdel, C. Elbuken, and C. Ren, "Passive droplet trafficking at microfluidic junctions under geometric and flow asymmetries," *Journal on Lab on a Chip*, vol. 11, no. 22, pp. 3774–3784, 2011.

[9] F. Jousse, R. Farr, D. R. Link, M. J. Fuerstman, and P. Garstecki, "Bifurcation of droplet flows within capillaries," *Physical Review E*, vol. 74, no. 3, p. 036311, 2006.

[10] B. J. Smith and D. P. Gaver III, "Agent-based simulations of complex droplet pattern formation in a two-branch microfluidic network," *Journal on Lab on a Chip*, vol. 10, no. 3, pp. 303–312, 2010.

[11] D. Sessoms, M. Belloul, W. Engl, M. Roche, L. Courbin, and P. Panizza, "Droplet motion in microfluidic networks: Hydrodynamic interactions and pressure-drop measurements," *Physical Review E*, vol. 80, no. 1, p. 016317, 2009.

[12] D. Sessoms, A. Amon, L. Courbin, and P. Panizza, "Complex dynamics of droplet traffic in a bifurcating microfluidic channel: Periodicity, multistability, and selection rules," *Physical Review Letters*, vol. 105, no. 15, p. 154501, 2010.

[13] M. D. Behzad, H. Seyed-Allaei, and M. R. Ejtehadi, "Simulation of droplet trains in microfluidic networks," *Physical Review E*, vol. 82, no. 3, p. 037303, 2010.

[14] O. Cybulski and P. Garstecki, "Dynamic memory in a microfluidic system of droplets traveling through a simple network of microchannels," *Journal on Lab on a Chip*, vol. 10, no. 4, pp. 484–493, 2010.

[15] Y.-C. Tan, J. S. Fisher, A. I. Lee, V. Cristini, and A. P. Lee, "Design of microfluidic channel geometries for the control of droplet volume, chemical concentration, and sorting," *Journal on Lab on a Chip*, vol. 4, no. 4, pp. 292–298, 2004.

[16] D. Link, S. L. Anna, D. Weitz, and H. Stone, "Geometrically mediated breakup of drops in microfluidic devices," *Physical Review Letters*, vol. 92, no. 5, p. 054503, 2004.

[17] J. Köhler, T. Henkel, A. Grodrian, T. Kirner, M. Roth, K. Martin, and J. Metze, "Digital reaction technology by micro segmented flow-components, concepts and applications," *Chemical Engineering Journal*, vol. 101, no. 1, pp. 201–216, 2004.

[18] Y.-C. Tan, Y. L. Ho, and A. P. Lee, "Droplet coalescence by geometrically mediated flow in microfluidic channels," *Journal of Microfluidics and Nanofluidics*, vol. 3, no. 4, pp. 495–499, 2007.

[19] W. Wang, C. Yang, and C. M. Li, "On-demand microfluidic droplet trapping and fusion for on-chip static droplet assays," *Journal on Lab on a Chip*, vol. 9, no. 11, pp. 1504–1506, 2009.

[20] A. Grimmer, W. Haselmayr, A. Springer, and R. Wille, "Design of application-specific architectures for Networked Labs-on-Chips," *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 193–202, 2018.

[21] L. Donvito, L. Galluccio, A. Lombardo, and G. Morabito, "Microfluidic networks: Design and simulation of pure hydrodynamic switching and medium access control," *Journal of Nano Communication Networks*, vol. 4, no. 4, pp. 164–171, 2013.

[22] M. J. Fuerstman, A. Lai, M. E. Thurlow, S. S. Shevkoplyas, H. A. Stone, and G. M. Whitesides, “The pressure drop along rectangular microchannels containing bubbles,” *Journal on Lab on a Chip*, vol. 7, no. 11, pp. 1479–1489, 2007.

[23] G. Cristobal, J.-P. Benoit, M. Joanicot, and A. Ajdari, “Microfluidic bypass for efficient passive regulation of droplet traffic at a junction,” *Applied Physics Letters*, vol. 89, no. 3, pp. 34 104–34 104, 2006.

[24] E. D. Leo, L. Donvito, L. Galluccio, A. Lombardo, G. Morabito, and L. M. Zanoli, “Design and assessment of a pure hydrodynamic microfluidic switch,” in *Int’l Conf. on Communications*, 2013, pp. 3165–3169.

[25] L. Donvito, L. Galluccio, A. Lombardo, and G. Morabito, “On the assessment of microfluidic switching capabilities in NLoC networks,” in *Int’l Conf. on Nanoscale Computing and Communication*, 2014, p. 19.

[26] ——, “μ-NET: a network for molecular biology applications in microfluidic chips,” *Trans. on Networking*, 2015.

[27] A. Grimmer, W. Haselmayr, A. Springer, and R. Wille, “Verification of Networked Labs-on-Chip architectures,” in *Design, Automation and Test in Europe*, 2017, pp. 1679–1684.

[28] ——, “A discrete model for Networked Labs-on-Chips: Linking the physical world to design automation,” in *Design Automation Conference*, 2017, pp. 50:1–50:6.

[29] R. Thakur, Y. Zhang, A. Amin, and S. Wereley, “Programmable microfluidic platform for spatiotemporal control over nanoliter droplets,” *Journal of Microfluidics and Nanofluidics*, vol. 18, no. 5-6, pp. 1425–1431, 2015.

[30] K. Churski, M. Nowacki, P. M. Korczyk, and P. Garstecki, “Simple modular systems for generation of droplets on demand,” *Journal on Lab on a Chip*, vol. 13, no. 18, pp. 3689–3697, 2013.

[31] S. A. Vanapalli, A. G. Banpurkar, D. van den Ende, M. H. Duits, and F. Mugele, “Hydrodynamic resistance of single confined moving drops in rectangular microchannels,” *Journal on Lab on a Chip*, vol. 9, no. 7, pp. 982–990, 2009.

[32] A. J. Teo, K.-H. H. Li, N.-T. Nguyen, W. Guo, N. Heere, H.-D. Xi, C.-W. Tsao, W. Li, and S. H. Tan, “Negative pressure induced droplet generation in a microfluidic flow-focusing device,” *Journal of Analytical Chemistry*, vol. 89, no. 8, pp. 4387–4391, 2017.

[33] M. Hamidović, W. Haselmayr, A. Grimmer, and R. Wille, “Towards droplet on demand for microfluidic networks,” in *Workshop on Molecular Communications*, 2018.

[34] S.-Y. Teh, R. Lin, L.-H. Hung, and A. P. Lee, “Droplet microfluidics,” *Journal on Lab on a Chip*, vol. 8, pp. 198–220, 2008.

[35] H. Gu, M. H. Duits, and F. Mugele, “Droplets formation and merging in two-phase flow microfluidics,” *Journal of Molecular Sciences*, vol. 12, no. 4, pp. 2572–2597, 2011.

[36] Y.-C. Tan, Y. L. Ho, and A. Lee, “Microfluidic sorting of droplets by size,” *Journal of Microfluidics and Nanofluidics*, vol. 4, no. 4, pp. 343–348, 2008.

[37] F. Jousse, G. Lian, R. Janes, and J. Melrose, “Compact model for multiphase liquid–liquid flows in micro-fluidic devices,” *Journal on Lab on a Chip*, vol. 5, no. 6, pp. 646–656, 2005.

[38] K. Song, G. Hu, X. Hu, R. Zhong, X. Wang, and B. Lin, “Encoding and controlling of two droplet trains in a microfluidic network with the loop-like structure,” *Journal of Microfluidics and Nanofluidics*, vol. 19, no. 6, p. 1363, 2015.

[39] X. Chen and C. L. Ren, “A microfluidic chip integrated with droplet generation, pairing, trapping, merging, mixing and releasing,” *Journal on RSC Advances*, vol. 7, no. 27, pp. 16 738–16 750, 2017.

[40] D. J. Acheson, *Elementary fluid dynamics*. Oxford University Press, 1990.

[41] P. Garstecki, M. J. Fuerstman, H. A. Stone, and G. M. Whitesides, “Formation of droplets and bubbles in a microfluidic T-junctionscaling and mechanism of break-up,” *Journal on Lab on a Chip*, vol. 6, no. 3, pp. 437–446, 2006.

[42] A. Biral and A. Zanella, “Introducing purely hydrodynamic networking functionalities into microfluidic systems,” *Journal of Nano Communication Networks*, vol. 4, no. 4, pp. 205–215, 2013.

[43] T. Thorsen, R. W. Roberts, F. H. Arnold, and S. R. Quake, “Dynamic pattern formation in a vesicle-generating microfluidic device,” *Physical Review Letters*, vol. 86, no. 18, p. 4163, 2001.

[44] A. Ajdari, “Steady flows in networks of microfluidic channels: building on the analogy with electrical circuits,” *Journal on Comptes Rendus Physique*, vol. 5, no. 5, pp. 539–546, 2004.

Andreas Grimmer received the Master’s degree in computer science from the Johannes Kepler University Linz, Austria, in 2015. From 2013 to 2015 he was a student researcher at the laboratory for Monitoring and Evolution of Very-Large-Scale Software Systems. Currently, he is a researcher at the Institute of Integrated Circuits and is working towards a PhD. His research focuses on methods for simulation and design automation for microfluidics.

Werner Haselmayr (S’08—M’13) received the Dipl.-Ing. degree in telematics from the Graz University of Technology, Austria and the Dr. techn. degree in mechatronics from the Johannes Kepler University Linz, Austria, in 2007 and 2013, respectively. He is currently an Assistant Professor at the Institute for Communications Engineering and RF-Systems at Johannes Kepler University. His research interests include algorithm design for wireless communications, iterative processing and molecular communication.

Rober Wille received the Diploma and Dr.-Ing. degrees in computer science from the University of Bremen, Germany, in 2006 and 2009, respectively. He has been with the Group of Computer Architecture, University of Bremen, Germany, from 2006–2015 and with the German Research Center for Artificial Intelligence (DFKI), Bremen, Germany, from 2013 onwards. Additionally, he worked as lecturer at the University of Applied Science of Bremen, Germany, and as Visiting Professor at the University of Potsdam, Germany, and the Technical University Dresden, Germany. Since 2015, he is Full Professor at the Johannes Kepler University Linz, Austria. His research interests are in the design of circuits and systems for both conventional and emerging technologies. In these areas, he published more than 200 papers in journals and conferences and served in editorial boards and program committees of numerous journals/conferences such as TCAD, ASP-DAC, DAC, DATE, and ICCAD.