

Automatic Design of Droplet-Based Microfluidic Ring Networks

Gerold Fink *Student Member, IEEE*, Medina Hamidović,
Werner Haselmayr *Member, IEEE*, and Robert Wille *Senior Member, IEEE*

Abstract—Droplet-based microfluidic networks allow to process biological or medical samples by standard unit operations such as mixing, incubating, sorting, or sensing. However, many of these networks usually perform such operations in a pre-defined way and, thus, lack in their flexibility. To overcome this problem, ring networks are used, since they allow to execute multiple operations in a row. But while several concepts and also prototypical implementations exist that realize such ring networks, the design process for them is still mainly conducted manually thus far. This is a severe drawback since various aspects such as the dimensions of the channels, the effects of droplets, the used fluids, the volumetric flow rates inside the channels, etc. have to be considered for this purpose. In this work, we propose design automation methods which address this problem. The proposed solution will automatically generate a proper design as well as correspondingly needed droplet sequences. A case study demonstrates the applicability of the resulting methods and simulations confirm the validity of the proposed approach.

Index Terms—droplet microfluidics, microfluidic networks, ring networks, design automation

I. INTRODUCTION

Microfluidic devices have been proposed in order to deal with the manipulation of small amounts of fluids [1]. They are used to realize experiments or operations in domains such as medicine, (bio-)chemistry, biology, pharmacology, etc., where tasks usually conducted in bulky and expensive laboratories should be minimized, integrated, and automated on a single device—often also called *Lab-on-a-Chip* (LoC) [2], [3]. *Droplet-based microfluidics* [4] provide a particular form of devices in which droplets of one fluid flow inside closed micro-channels and are transported by a second immiscible fluid, which acts as a carrier fluid for the droplets. This allows to process biological or medical samples (encapsulated within a so-called *payload droplet*) by routing this payload droplet through different *modules* that realize standard unit operations such as mixing, incubating, sorting, or sensing. This concept found great applications, e.g., for DNA sequencing, cell analysis, organism analysis, and drug screening [4].

However, most droplet-based microfluidic devices process the available operations in a pre-defined way [4]. This obviously limits its flexibility. Hence, in order to allow for a more flexible use, the concept of microfluidic networking has been proposed in [5]—leading to *droplet-based microfluidic networks*. This allows to route the payload droplet through different sequences of modules in an order which is not pre-defined but rather can be adjusted for a particular application and/or experiment. Moreover, only passive hydrodynamic effects are utilized in such droplet-based microfluidic networks, i.e., no

active and error-prone components such as valves are needed. Following this scheme, the design and realization of several microfluidic topologies have been investigated in the recent past.

For example, in its most specific form, application-specific networks have been proposed in [6], where the corresponding network topology is adjusted to a pre-defined (and application-specific) set of experiments. Although corresponding design methods for that are already available (see, e.g., [7], [8]), physical realizations for those networks are still in its infancy. Furthermore, application-specific networks are, by definition, focused on dedicated applications and are not suitable if a payload droplet should be processed by various different sequences of modules. Also other topologies such as bus networks [9], [10] would be possible. However, since each output channel of a bus network leads only to a single module, the payload droplet always has to be re-injected into the network again, once it was processed by a module – a rather crucial and erroneous step. Therefore, bus networks are typically used when only a single module should be addressed.

In fact, in order to realize a network which is flexible and allows the payload droplet to get processed by different sequences of modules, a *ring network* as originally introduced in [5], [11] is a more suitable topology. Here, (1) multiple nodes are connected in series, (2) each node contains a module realizing a different operation, and (3) a switching mechanism (based on passive hydrodynamic effects) controls whether the payload is processed by the module of that node or not. This eventually allows to execute arbitrary sequences of operations.

Moreover, the ring network also belongs to the most advanced topologies with respect to its conceptual and technological maturity. In fact, while the original concepts proposed in [5] only covered the execution of a single module per droplet injection, the solution proposed in [12] extended this concept to address multiple modules—significantly improving the performance. However, the approach in [12] uses a very complex droplet-by-size sorter, which needs different droplet sizes in order to work as expected. While precise but still distinct droplet volumes are rather hard to produce during the droplet injection, the sorter is also much more sensitive and error-prone; as compared to a simple T-Junction with bypass channel used in the approach presented in this work. Furthermore, sophisticated solutions for simulating droplet-based microfluidic networks [13], [14], [15] are available which confirmed the plausibility and general functionality of the ring network. Finally, also schemes for the prototypical implementations of the respectively required switching

mechanisms [16], [17] and droplet-on-demand generation [18], [19] (needed to inject droplets with a particular pre-defined distance) have been proposed.

However, despite these accomplishments, the design process for droplet-based microfluidic ring networks is still mainly conducted manually thus far. This is a severe drawback since the respective design tasks are highly non-trivial. In fact, to route a payload droplet so that only the intended modules are executed, further droplets called *header droplets* are needed. These droplets do not contain any biological/chemical sample and are only used for routing purposes. But their routing behavior significantly depends, e.g., on their distance to the payload droplet and the sizes of corresponding channels within the network. Determining a proper design together with corresponding sequences of payload and header droplets requires the consideration of various aspects such as the dimensions of the channels, the effects of droplets, the used fluids, the volumetric flow rates inside the channels, etc.—infeasible to handle manually anymore.

In this work, we close this gap by providing automatic design methods for droplet-based microfluidic ring networks. To this end, we first define the considered problem and sketch the general idea of the ring network in Section II. Then, a comprehensive description of the proposed solutions is provided in Section III. This includes a review of the physical model which is used to derive corresponding design conditions that need to be satisfied in order to realize the desired behavior. Based on that, methods are proposed that automatically generate designs of droplet-based microfluidic ring networks as well as corresponding droplet sequences realizing arbitrary experiments on those networks. All resulting methods have been evaluated and validated using state-of-the-art simulation engines. The results of those evaluations are reported in Section IV. By this, first comprehensive solutions for the design automation of droplet-based microfluidic ring networks become available.

II. MOTIVATION

This section describes and illustrates the considered problem in more detail and provides the general idea of the proposed solution.

A. Considered Problem

Microfluidic networks allow to process so-called *payload* droplets (i.e., droplets containing a biological sample) through different components (called *modules* in the following) realizing standard unit operations like mixing, incubating, sorting, or sensing. In order to allow for a flexible use of the device and its components, the network should support the execution of different sequences of modules—defining different experiments. More formally, given a set of modules $M = \{m_1, m_2, \dots, m_N\}$, different experiments $E = M^n = M \times \dots \times M = \{(m^{(1)}, \dots, m^{(n)}) | m^{(i)} \in M, i = 1, \dots, n\}$ shall be realized through the microfluidic network.

Example 1. *Let's assume that realizations of modules $M = \{m_1, m_2, \dots, m_6\}$ are available.*

Then, possible experiments to realize through a microfluidic network could be $E_1 = (m_2, m_3, m_5, m_6)$ or $E_2 = (m_1, m_2, m_6, m_1, m_4, m_5, m_2)$.

In order to realize such experiments, a variety of different network topologies can be used (e.g., bus, star, and ring networks). Bus and star networks have usually one input channel and multiple output channels, where each output channel leads to a single module. Hence, once the payload droplet is routed towards such a module and gets processed, it must be re-injected into the input channel of the network again in order to address the next module. Unfortunately, such re-injections are very crucial and erroneous steps which can even lead to the destruction of the sample inside the payload droplet. As a result, bus and star networks are not suited when mainly experiments are considered, where multiple modules should be addressed in a row. In fact, a much more appropriate choice for such experiments are ring networks [5], [11], since their modules are arranged in series and, thus, allow to address multiple modules without re-injecting the payload droplet after it was processed by a single module. Therefore, such ring networks are considered in the following¹.

The basic structure of a ring network is shown in Fig. 1, where the control unit is responsible for the injection of droplets into the network as well as for their removal. Moreover, the ring network consists of N nodes, which are connected in series and where each node contains a different module $m_i \in M$. A payload droplet which is injected into the network flows through all nodes, until it reaches the control unit again. In each node and depending on the correspondingly considered experiment, the payload droplet should either be processed by the module m_i of this node or should pass the node unchanged. Therefore, an effective realization of a node is needed, which allows to control the payload droplet accordingly. Since additionally a droplet can only flow into one direction (i.e., from Node₁ towards Node _{N}), this leads to the following two approaches to realize a particular experiment:

- *Single-Node Approach*, i.e., only the module of one node is addressed per injection process [5].
- *Multi-Node Approach*, i.e., modules of multiple nodes are addressed per injection process [12].

As a result, the payload droplet may have to be injected several times into the network, depending on the particular experiment and if a single-node or multi-node approach is used.

Example 2. *Consider the two experiments E_1 and E_2 introduced in Example 1. Following the single-node approach, a total of four and seven injection cycles are needed for E_1 and E_2 , respectively. Following the multi-node approach, E_1 can be realized with a single cycle, while E_2 requires a total of three cycles (the first one executing m_1, m_2, m_6 followed by another one executing m_1, m_4, m_5 and the final one executing m_2).*

When multiple modules should be addressed in a row and when a high flexibility of the used modules is desired, a ring network provides an appropriate solution in order to

¹Note that the solutions proposed here can, in general, also be applied to other topologies.

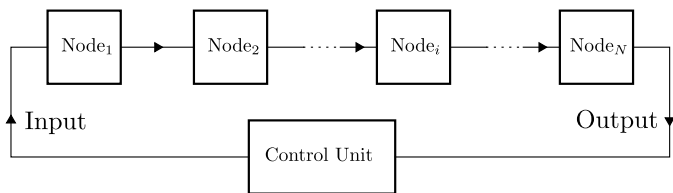


Fig. 1: Structure of a ring network.

address these desires. However, how to efficiently design and realize such microfluidic networks has not been addressed yet, because the addressing process of the nodes (especially for the multi-node approach) as well as their design are non-trivial problems. In this work, we investigate these problems and propose automatic solutions for them.

B. General Idea

In order to realize a ring network with the abilities described in the previous section, a suited concept for the nodes has to be established first. Since the nodes should be as simple as possible, no active mechanisms like valves should be used and only passive hydrodynamic effects should be utilized.

As mentioned before, a node must have the ability to route a payload droplet through itself, without directing the droplet through the module. As a result, it is essential for a node to have at least two different paths, where only one of them contains the module, i.e., a *module path* and a *non-module path* (as sketched in Fig. 2a). Furthermore, it is necessary for the node to have some kind of switching mechanism which controls whether the payload droplet is either routed into the module path or into the non-module path. These two requirements can be achieved by the microfluidic design shown in Fig. 2b.

As requested, the design connects the input and output channel of the node by a module path (channels c2 and c5) and a non-module path (channels c1 and c4). Moreover, the switching mechanism is accomplished by exploiting only passive hydrodynamic effects and is realized with a bifurcation (channels c1 and c2) and an additional bypass channel c3. This switching mechanism is based on the concepts described e.g., in [16], [17] and works as follows:

Each channel of the network has a certain hydrodynamic resistance, which mainly depends on the channel geometry (i.e., the smaller the diameter and/or the longer the channel, the higher the resistance) and the viscosity of the continuous phase, i.e., the phase which acts as a carrier fluid for the droplets. As indicated in Fig. 2b, the length of channel c1 is shorter than the length of channel c2, resulting in a smaller hydrodynamic resistance of channel c1 compared to channel c2. The bypass channel c3 is used to make the switching process only dependent on the resistances of the two channels c1 and c2 and independent of the subsequent channels c4 and c5. Furthermore, the bypass channel c3 cannot be entered by any droplet and only lets the continuous phase flow through it (cf. barriers in Fig. 2b).

Having this setup, a droplet which reaches the bifurcation point *B* will always enter the channel with the lowest hydrodynamic resistance (in other words, the channel with the

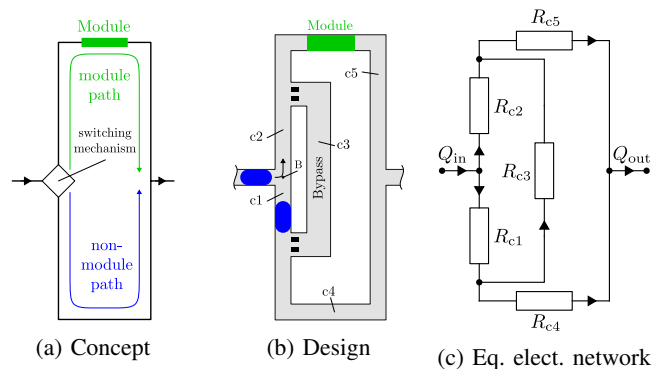


Fig. 2: Realization of a node.

highest volumetric flow rate), i.e., channel c1 in this case. This behavior makes the channel c1, i.e., the non-module path, the default path. Therefore, a single payload droplet which flows through the ring network will always enter the default path, i.e., the non-module path, of each node and never gets processed by any module.

However, in order to route the payload droplet into the module path, passive hydrodynamic effects can be utilized as well. This is, because a droplet increases the hydrodynamic resistance of the channel in which it is contained through its viscosity and volume (as studied in [20]). Hence, by injecting an additional droplet into the node before the payload droplet, it can be arranged that the hydrodynamic resistance of channel c1 becomes larger than the resistance of channel c2—eventually making the payload droplet entering the module path. Those additional droplets are called *header droplets*, do not contain any biological samples, and are only used to route the payload droplet into the desired module. Overall, this allows, in principle, for a realization of a microfluidic ring network as discussed in the previous section.

III. AUTOMATIC DESIGN OF RING NETWORKS

The general idea sketched above is an obvious way to realize a microfluidic device that allows the execution of different sequences of modules. However, in order to route the payload droplet along the module path (or the non-module path), a header droplet must occupy (must not occupy) channel c1 exactly at the time, when the payload droplet reaches the bifurcation point *B*. That is, the distances between the respective droplets play an important role and need to be properly defined so that the payload is indeed routed as intended. Additionally, those distances heavily depend on the precise dimensions (e.g., lengths) of the respectively involved channels. On top of that, all that does not only need to be considered for a single node, but throughout the entire network, e.g., the droplet distance must be small enough in one node (to get the module of this node executed), but large enough in another node (to avoid the execution of the module of this node). Overall, this yields a highly non-trivial design task which hardly can be addressed manually.

In this section, we propose an automatic method that addresses this task and automatically generates precise designs (i.e., precise dimensionings for different sets of modules) as

well as droplet sequences (with proper distances between droplets so that the payload is indeed routed through the desired modules). To this end, we first provide

- a physical model that allows to describe the behavior of microfluidic systems in general,
- a definition of conditions for the channel dimensions that need to be satisfied in order to realize the desired behavior, and
- a definition of conditions for the droplet distances that need to be satisfied in order to realize the desired behavior.

Afterwards, we describe how the desired routing behavior can be automatically realized for both, the single-node approach and the multi-node approach.

A. Physical Model

The proposed solution requires a formal basis describing the behavior of microfluidic systems. To this end, the one-dimensional (1D) analysis model proposed in [21] can be utilized. This model assumes that pumps produce a fully developed and laminar flow (usually at low Reynolds numbers)—a property which is clearly satisfied in the case considered here. Then, the flow inside a channel can be described by Hagen-Poiseuille's law [22], [23]

$$\Delta P = Q \cdot R, \quad (1)$$

where Q is the volumetric flow rate, ΔP the pressure drop along the channel, and R the hydrodynamic resistance of the channel. This hydrodynamic resistance depends on the channel geometry (i.e., length l , width w , and height h) as well as the dynamic viscosity of the continuous phase μ_c . More precisely, for rectangular channels with a section ratio $h/w < 1$, the hydrodynamic resistance can be determined by

$$R = \frac{a \mu_c l}{w h^3}, \quad (2)$$

where a denotes a dimensionless parameter defined as

$$a = 12 \left[1 - \frac{192 h}{\pi^5 w} \tanh\left(\frac{\pi w}{2h}\right) \right]^{-1}. \quad (3)$$

As already mentioned in Section II-B, droplets also increase the resistance of a channel. As proposed in [20], a droplet with the length l_{droplet} increases the resistance of the segment it occupies inside the channel by 2 – 5 times. In this work, the factor 3 is assumed, i.e.,

$$R_{\text{droplet}} = \frac{3 a \mu_c l_{\text{droplet}}}{w h^3}. \quad (4)$$

Please note that designers can always adjust this factor in order to meet their particular requirements of their experimental settings, when the used factor and the hydrodynamic droplet resistance of an actual physical realization do not properly match.

This 1D analysis model can now be used to describe the behavior of a microfluidic network. In order to realize this, the channels of such networks can be represented by their hydrodynamic resistances, which leads to the so-called equivalent electrical network [23]. Note that the term of electrical

network is used, because the representation is equivalent to an electrical resistor network. The equivalent electrical network of a node (cf. Fig. 2b) is shown in Fig. 2c and can be used to compute the volumetric flow rates inside the channels. Please note that, for sake of simplicity, the module is also represented as a simple channel and its resistance is already included into the resistance of channel c5.

B. Conditions for Channel Dimensions

The 1D analysis model provides the basis for formulating the conditions that need to be satisfied in order to realize the behavior sketched in Section II-B. In fact, certain flow rates need to be realized to route a droplet by default to the non-module path and, with an additional header droplet, to the module path. Thanks to the 1D-model, corresponding formulations are available. More precisely, the *normalized* volumetric flow rate of the channel c1 (which describes the flow rate of that channel normalized to the flow rate of the input channel Q_{input}) can be determined by

$$q_{c1} = \frac{Q_{c1}}{Q_{\text{input}}} = \frac{R_{c2} + R_{\beta}}{R_{c1} + R_{\alpha} + R_{c2} + R_{\beta}}, \quad (5)$$

where the hydrodynamic resistances R_{α} and R_{β} are obtained with the help of a delta-star-transformation of the resistances R_{c3} , R_{c4} , and R_{c5} (cf. Fig. 2c), and yield

$$R_{\alpha} = \frac{R_{c3} R_{c4}}{R_{c3} + R_{c4} + R_{c5}}, \quad R_{\beta} = \frac{R_{c3} R_{c5}}{R_{c3} + R_{c4} + R_{c5}}. \quad (6)$$

A single droplet which reaches the node should be routed into the non-module path, i.e., into channel c1. Since a droplet always flows inside the channel with the lowest hydrodynamic resistance, or in other words, with the highest instantaneous flow rate, the flow rate of channel c1 has to be higher than the flow rate of channel c2. Therefore, the first condition to be satisfied by a design realizing the desired behavior is

$$q_{c1} > q_{c2} \xrightarrow{q_{c1} + q_{c2} = 1} q_{c1} > 0.5. \quad (7)$$

That is, q_{c1} has to be greater than q_{c2} . Since the equation $Q_{c1} + Q_{c2} = Q_{\text{in}}$ and, hence, also the equation $q_{c1} + q_{c2} = 1$ hold (due to Kirchhoff's current law), this means that the normalized volumetric flow rate q_{c1} for channel c1 needs to be greater than 0.5.

Furthermore, a second droplet, i.e., the payload droplet, closely following the first one, should be routed into the module path, i.e., into channel c2. This implies that, when a droplet is present inside c1, the flow rate of channel c2 has to be larger than the flow rate of channel c1—yielding the second condition to be satisfied, namely

$$q_{c1}^* < q_{c2}^* \xrightarrow{q_{c1}^* + q_{c2}^* = 1} q_{c1}^* < 0.5, \quad (8)$$

where the *-sign indicates that a droplet is present inside the channel c1, i.e., the resistance of c1 has to be substituted with $R_{c1} \rightarrow R_{c1} + R_{\text{droplet}}$, when q_{c1} is computed.

Finally, another issue which has to be considered is that the module path has to be longer (by a certain value) than the non-module path. This is because, once a header droplet has triggered the switching mechanism of a node and routed

the payload droplet into the module path (cf. Fig. 3a), the header droplet must not trigger the switching mechanism of a subsequent node. Hence, the header droplet must pass the node much faster than the payload droplet, so that the header droplet is able to “escape” the payload droplet and does not trigger the switching mechanism of an undesired node. This is important, because otherwise every time this particular node gets addressed it would unavoidably lead to the case that also the corresponding subsequent node gets addressed – resulting in an undesired behavior, when the payload droplet should not be processed by the module of the subsequent node (which is certainly the case in several experiments). Therefore, the module path has to be designed in such a way that it is longer than the non-module path, like it is indicated in Fig. 3a. More precisely, the time t_m the payload droplet needs to pass a node through the module path should be at least twice as much as the time t_{nm} the header droplet needs to pass the node through the non-module path, i.e., $t_m = b \cdot t_{nm}$ with $b > 2$. With this assumption, it is ensured that the header droplet already passed the subsequent node, before the payload droplet arrives at it. As a result, the two droplets cannot influence each other anymore and, thus, it is not possible that the switching mechanism of a subsequent node gets triggered. In other words, the header droplet is able to pass two nodes in the same time as the payload droplet passes a single node, which is illustrated by Fig. 3b. These times can be approximated by the following two equations

$$t_m \approx \frac{l_m A}{q_m Q_{\text{input}}} \quad t_{nm} \approx \frac{l_{nm} A}{q_{nm} Q_{\text{input}}}, \quad (9)$$

where A is the cross-section of the channels, Q_{input} the volumetric flow rate of the input channel, and where l_m and l_{nm} are the lengths of the module and non-module paths, respectively. Moreover, q_m and q_{nm} are the normalized volumetric flow rates of the module respectively non-module paths which can also be approximated by

$$q_m \approx \frac{l_{nm}}{l_m + l_{nm}} \quad q_{nm} \approx \frac{l_m}{l_m + l_{nm}}. \quad (10)$$

Inserting these equations into $t_m = b \cdot t_{nm}$ yields

$$l_m^2 = b l_{nm}^2, \quad (11)$$

which can then be reorganized with $b > 2$ to obtain the following condition

$$b = \left(\frac{l_m}{l_{nm}} \right)^2 > 2. \quad (12)$$

However, this approach only works properly, when all nodes in the ring network share the same length for the module respectively non-module path. More precisely, the length $l_m = l_{c2} + l_{c5}$ must be identically throughout all nodes as well as the length $l_{nm} = l_{c1} + l_{c4}$.

Overall, this leads to conditions in terms of Eqs. (7), (8), and (12) which, if satisfied, yield a valid realization of the nodes employing the desired behavior. Moreover, these conditions allow for an automatic generation of nodes (and, hence, the ring network), by solving simple equation systems

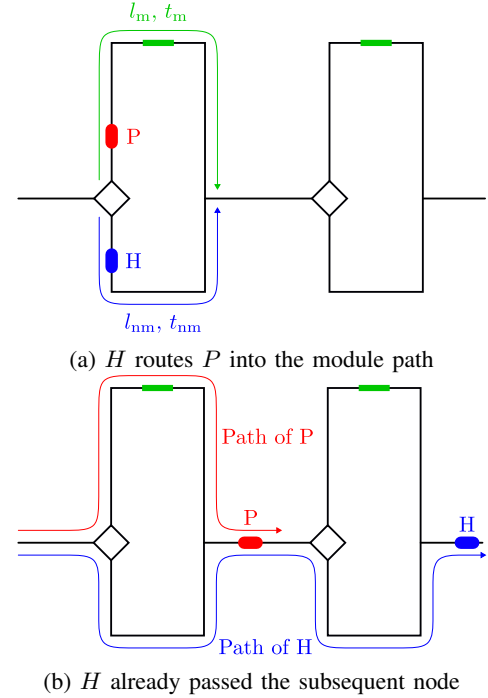


Fig. 3: A header (H) and payload droplet (P) pass a node.

which are based on these conditions and just require some certain predefined parameters such as the channel widths and heights, the viscosity of the continuous phase, etc.

C. Conditions for Droplet Distances

Once a ring network satisfying the conditions from above, i.e., Eqs. (7), (8), and (12), has been realized, the payload droplet will by default enter the non-module path of a node. The module path will be entered when a header droplet precedes the payload droplet. However, this only works, if channel $c1$ is still occupied by the header droplet, when the payload droplet reaches the bifurcation point B (cf. Fig. 2b). As already mentioned in Section II-B, this can be controlled by the distances between the two droplets.

More precisely, if the payload is supposed to be executed by a module (denoted by ON in the following), the distance between the header and the payload $d_{H,P}$ when arriving at a node should be shorter than a certain threshold τ_{ON} (cf. Fig. 4a). In contrast, if the payload is *not* supposed to be executed by a module (denoted by OFF in the following), this distance needs to be longer than a certain threshold τ_{OFF} (cf. Fig. 4b). These two thresholds can be determined by the equations²

$$\tau_{ON} = \frac{1}{q_{c1}^*} (l_{c1} - l_{\text{droplet}}) + l_{\text{droplet}} \quad \text{and} \quad (13)$$

$$\tau_{OFF} = \frac{1}{q_{c1}^*} l_{c1} + l_{\text{droplet}}, \quad (14)$$

where l_{c1} and l_{droplet} are the lengths of the channel $c1$ and the droplet, respectively. Hence, when injecting the respective

²Note that the condition $\tau_{ON} < \tau_{OFF}$ is always satisfied for a single node.

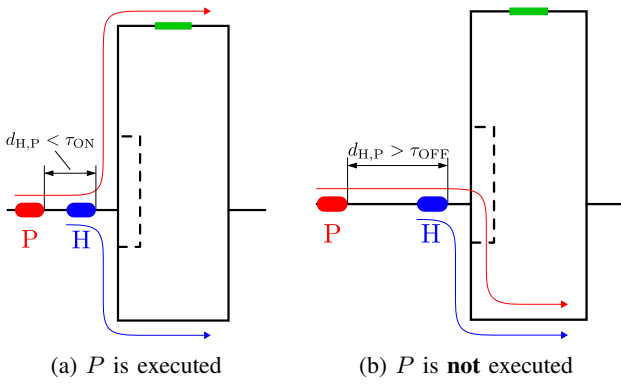


Fig. 4: Executing a payload droplet (P) by a module.

droplets, it should be made sure that, when both droplets eventually arrive at the node, their distance is below or above those thresholds. Note that, if the actual droplet distance is in between τ_{ON} and τ_{OFF} , no safe statement about the path of the payload droplet can be made and, therefore, such distances should be avoided.

Overall, the payload droplet can be routed into the module path or non-module path, when the distance between the header droplet H and the payload droplet P satisfies $d_{H,P} < \tau_{ON}$ or $d_{H,P} > \tau_{OFF}$, respectively. This can be used to generate droplet sequences with proper distances that eventually execute the desired experiment.

D. Generalization for Single-Node Approach

The conditions defined above describe the dimensioning and distances needed to realize the desired behavior with respect to a single node only. In a next step, they now have to be generalized for an entire network composed of several nodes. In the following, this is first described for the single-node approach, where only the module of one node is executed per injection process as illustrated in Example 2. Therefore, e.g., the experiment $E_1 = (m_2, m_3, m_5, m_6)$ must be split into four sub-experiments, i.e., $E_{1,1} = (m_2)$, $E_{1,2} = (m_3)$, $E_{1,3} = (m_5)$, and $E_{1,4} = (m_6)$. In each of these experiments, a particular module respectively node (denoted by the index i in the following) has to be addressed.

At a first glance, the realization of the single-node approach looks simple. The network simply has to be dimensioned and realized so that the conditions stated in Eqs. (7), (8), and (12) are satisfied. Then, a header droplet H , and a payload droplet P have to be injected so that they eventually arrive the i^{th} node with a distance of $d_{H,P}$ (as sketched in Fig. 6a) that satisfies the condition $d_{H,P} < \tau_{ON}$. However, a problem is that, if both droplets always take the same path, i.e., the non-module path, the distances between them before and after a node are nearly identical. When now all nodes would share the same channel dimensions, the thresholds τ_{ON} and τ_{OFF} would also be identical for all nodes. This would lead to the effect that solely the first node can be addressed, while all other nodes would not be addressable.

In order to overcome this problem, the thresholds τ_{ON} and τ_{OFF} have to be different for each node. Accordingly,

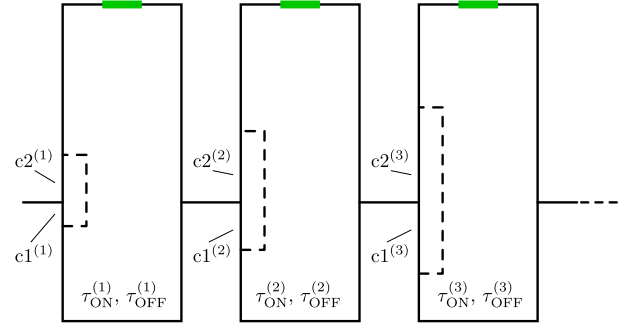


Fig. 5: Nodes with increasing dimensions for $c1$ and $c2$

the dimensions of the channels of each node (in particular the channels $c1$ and $c2$ of each node) have to be different (cf. Fig. 5). More precisely, the channel $c1$ of the i^{th} node needs to be longer than the channel $c1$ of the $(i-1)^{\text{th}}$ node (since additionally the conditions stated in Eq. (7) and Eq. (8) still have to hold, this also requires longer dimensions for channel $c2$). The precise difference in dimension is determined through the thresholds of each node, i.e., the equation

$$\tau_{ON}^{(1)} < \tau_{OFF}^{(1)} < \tau_{ON}^{(2)} < \tau_{OFF}^{(2)} < \dots < \tau_{ON}^{(N)} < \tau_{OFF}^{(N)} \quad (15)$$

has to be satisfied, where the superscript indicates the corresponding node in the network (cf. Fig. 1). By considering Eq. (15) and the design conditions from Section III-B, i.e., Eqs. (7), (8), and (12), a realization of a ring network which satisfies these conditions can be easily obtained. Again, this can be done in an automatic fashion, where simple equation systems, based on these conditions, have to be solved.

When such a realization of a ring network is obtained, the next step is to determine the corresponding distance $d_{H,P}$ between the two droplets. If the i^{th} node should be addressed, this distance must satisfy the condition $\tau_{OFF}^{(i-1)} < d_{H,P} < \tau_{ON}^{(i)}$ in front of the i^{th} node. However, the thresholds τ_{ON} and τ_{OFF} are upper respectively lower bounds which should not be exceeded and, thus, it would be less error-prone, if the droplet distance $d_{H,P}$ has the value

$$d_{\text{Process}}^{(i)} = \frac{\tau_{ON}^{(i)} + \tau_{OFF}^{(i-1)}}{2} \quad (16)$$

in front of the i^{th} node³.

Since the droplet distance stays nearly the same, when both droplets flow through the non-module path of a node, it is possible to inject the droplets exactly with this distance into the input channel of the ring network and, therefore, only the switching mechanism of the i^{th} node is triggered.

Overall, if the module m_i should be addressed, a payload and a header droplet have to be injected into the ring network which have a droplet distance of $d_{H,P} = d_{\text{Process}}^{(i)}$ defined by Eq. (16).

³When the first node ($i = 1$) should be addressed, the distance can be simply set to $d_{\text{Process}}^{(1)} = \tau_{ON}^{(1)}$, since $\tau_{OFF}^{(0)}$ does not exist.

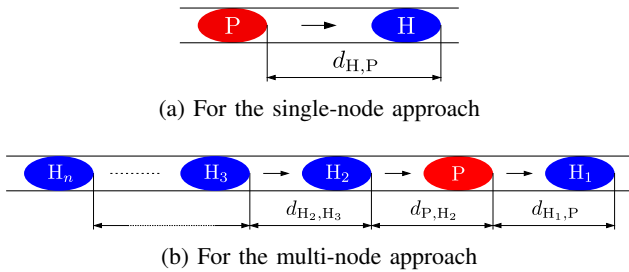


Fig. 6: Distances between payload and header droplets.

E. Generalization for Multi-Node Approach

In the previously described single-node approach, a payload droplet can only be processed by one module per injection process and has to be re-injected if more than one module should be addressed. However, a re-injection of the payload droplet is a rather difficult process—especially when the sample in the payload droplet is very sensitive. Therefore, such re-injections have a big potential of failures, introduce an additional overhead, and, hence, should be avoided.

In order to address this problem, a multi-node approach is presented in this section, which can route the payload droplet into more than one module per injection process and, thus, reduces the re-injections of the payload droplet. To this end, e.g., the experiment $E_2 = (m_1, m_2, m_6, m_1, m_4, m_5, m_2)$ is split into three sub-experiments, i.e., $E_{2,1} = (m_1, m_2, m_6)$, $E_{2,2} = (m_1, m_4, m_5)$, and $E_{2,3} = (m_2)$ when using the multi-node approach (as already discussed in Example 2). In the following, only single sub-experiments are considered—the overall experiment can be realized by combining the sub-experiments. In such a sub-experiment, the modules are sorted according to their appearance in the nodes and only one module per sub-experiment is allowed, i.e., $E' = (m_{i_1}, m_{i_2}, \dots, m_{i_n})$ with $i_1, i_2, \dots, i_n \in \{1, 2, \dots, N\}$ and $i_1 < i_2 < \dots < i_n$, where $n \leq N$ is the number of modules which should be addressed in the sub-experiment.

Furthermore, also for the multi-node approach, the channel dimensions of each node have to be different in order to allow to distinctly address each module. Hence, the design of the ring network is the same as in the single-node approach, i.e., Eqs. (7), (8), (12), and also (15) are all satisfied.

Having that, all what is left is, again, a proper definition of the required droplet sequence with proper distances between payload and header droplets. Since now multiple modules are addressed, also multiple header droplets have to be injected along with the payload droplet—as shown in Fig. 6b, where the payload droplet P (red) is always at the second position regardless which nodes or how many of them should be addressed. Each header droplet is then used to trigger the switching mechanism at the desired node and to route the payload droplet into the corresponding module, e.g., the header droplet H_1 is responsible to route the payload droplet into module m_{i_1} , the header droplet H_2 is responsible to route the payload droplet into module m_{i_2} , and so on.

The addressing process of the desired nodes can then be described as follows: Similar to the single-node approach, the first header droplet H_1 triggers the switching mechanism of

the first node which should be addressed and, thus, routes the payload droplet into the module path towards the desired module. Because the module path of a node is much longer than the non-module path (see Section III-B), the header droplet passes the node much faster than the payload droplet and, therefore, does not influence the payload droplet anymore. In other words, this header droplet cannot be used to trigger the switching mechanism of another node anymore.

Hence, a second header droplet H_2 is injected into the network if the payload droplet should be routed through the module of a second node. Since the payload droplet needs more time inside the module path, the second header droplet is able to catch up the distance d_{P,H_2} they had at injection time (cf. Fig. 6b) because it solely flows through the (faster) non-module paths of the nodes. Therefore, it is possible that this header droplet positions itself right in front of the payload droplet when both droplets leave the node which was just addressed. The distance they have after this node determines now which node is addressed next. In fact, the second header droplet H_2 and the payload droplet P have now the same constellation as the header droplet H_1 and the payload droplet at injection time. In the same way, this scheme can be applied for every additional node which should be addressed and where an additional header droplet is required.

However, in order to address the right nodes, the distance between the different droplets at injection time (cf. Fig. 6b) have to be obtained. Because the two droplets cannot influence each other anymore after the switching mechanism is triggered, this allows to determine the droplet distances at the input channel of the ring network one after another. To this end, the distance $d_{H_1,P}$ is computed first, which can be achieved exactly as for the single-node approach, and is then considered a fix value. When $d_{H_1,P}$ is fix, it is possible to determine the second distance, i.e., $d_{H_2,P}$, which is then also considered a fix value. By applying this scheme on the remaining distances, all required droplet distances at the input channel can be obtained.

However, besides the determination of the distance $d_{H_1,P}$, the computation of all other distances is non-trivial and a broad variety of aspects have to be taken into consideration, e.g., the actual channel dimensions of each node as well as which nodes and how many of them were already addressed. While all these aspects eventually could be determined with great effort by utilizing the 1D model described in Section III-A, using simulation engines such as [13], [14], [15] is the more suitable way (mainly because those simulators for droplet-based microfluidic networks are rather efficient nowadays and allow for an easy consideration of all relevant aspects).

Then, the problem to determine a specific droplet distance can be described as follows: Determine the droplet distance⁴ d_{H_{j-1},H_j} with $j \in \{3, \dots, n\}$ at the input channel of the network, so that the droplet distance before the j^{th} node which should be addressed matches $d_{\text{Process}}^{(j)}$ of Eq. (16).

Overall, this leads to the following procedure of determining the droplet distance d_{H_{j-1},H_j} (again, this can also be applied for the distance d_{P,H_2}):

⁴Please note that this description is also valid for the distance d_{P,H_2}

- 1) Form the droplet frame shown in Fig. 6b, but only consider the droplets H_1, P, H_2, \dots, H_j .
- 2) Set the droplet distances, i.e. $d_{H_1,P}, d_{P,H_2}, d_{H_2,H_3}, \dots, d_{H_{j-2},H_{j-1}}$, accordingly to the already known values (obtained from previous computations of the droplet distances). For the desired distance d_{H_{j-1},H_j} approximate a start value, e.g., by $d_{\text{start}} = \tau_{\text{OFF}}^{(j)}$.
- 3) Simulate the whole ring network assuming the droplet sequence resulting from the prepared droplet frame.
- 4) Compare the distance between the payload P and header droplet H_j in front of the j^{th} node with the required distance $d_{\text{Process}}^{(j)}$. If the two distances match, the desired droplet distance d_{H_{j-1},H_j} at the input channel has been obtained. Otherwise, adapt the distance d_{H_{j-1},H_j} at the input channel appropriately, and continue with Step (3).

Repeating this process for all header droplets H_j with $j \in \{3, \dots, n\}$ eventually yields all distances to execute the (sub-)experiment.

IV. APPLICATION AND VALIDATION

The methods described above have been implemented in Java and resulted in a tool which automatically generates designs of droplet-based microfluidic ring networks incorporating an arbitrary number of modules as well as corresponding droplet sequences realizing arbitrary experiments on those networks (both, following the single- and multi-node approach). In this section, we demonstrate the applicability and validity of the resulting tools by means of case studies. To this end, we first discuss the generation of the network design, followed by the determination of the correct droplet sequences for particular (sub-)experiments. Afterwards, the obtained results and their validity is discussed. In order to compute the results, we utilized a PC system with an Intel Core i5-8250 processor with 1.6 GHz and 8 GB of main memory.

A. Generating the Design

First, the design of the ring network is generated. To this end, the conditions proposed in Section III-B and Section III-D are applied. Fig. 7a summarizes the correspondingly needed steps. The first step is to provide the initial parameters such as the channel height and width, the viscosity of the continuous phase, etc., which depend on the respectively used setup. In our case study, we defined these initial parameters by the following values: The channel height $h = 33\mu\text{m}$, the width of the bypass channel c_3 $w_{c_3} = 200\mu\text{m}$, the width of all other channels $w = 100\mu\text{m}$, the droplet length $l_{\text{droplet}} = 150\mu\text{m}$ (inside a channel width the cross-section $A = w \cdot h$), the dynamic viscosity of the continuous phase $\mu_c = 1\text{mPas}$ and, finally, the volumetric flow rate of the pump $Q_{\text{input}} = 3 \cdot 10^{-11} \frac{\text{m}^3}{\text{s}}$. Besides these initial parameters, additional values have to be defined, which are important for the actual design of the ring network, such as the number of modules $N = 6$, the length of the module path $l_m = l_{c_2} + l_{c_5} = 25\,000\mu\text{m}$ and the length of the non-module path $l_{nm} = l_{c_1} + l_{c_4} = 15\,000\mu\text{m}$.

Using these inputs, the lengths of the different channels can be automatically determined as described in Section III-B and

Generating the Design:

- Define the initial parameters
 - Dynamic viscosity of the continuous phase
 - Channel width and height
 - Droplet length
 - Volumetric flow rate of the pump
 - Amount of modules
 - Lengths of module and non-module path
- Determine node dimensions, while considering the corresponding conditions
 - For each node: Eqs. (7), (8), (12)
 - For all nodes at once: Eq. (15)

(a) Generating the design

Generating Droplet Sequences:

- Define desired (sub-)experiments
- Compute the required droplet distances in front of the corresponding nodes, by using Eq. (16)
- Determine the droplet distances at the input channel by applying the procedure described in Section III-E

(b) Generating droplet sequences

Fig. 7: Steps to be conducted.

TABLE I: Resulting specification of the ring network.

	Nodes					
	Node ₁	Node ₂	Node ₃	Node ₄	Node ₅	Node ₆
Channel Lengths: all lengths are given in μm						
l_{c_1}	469	743	1021	1300	1580	1862
l_{c_2}	531	757	979	1200	1420	1638
l_{c_3}	1200	1700	2200	2700	3200	3700
l_{c_4}	14531	14257	13979	13700	13420	13138
l_{c_5}	24469	24243	24021	23800	23580	23362
Design Parameters: τ_{ON} and τ_{OFF} are given in μm						
τ_{ON}	881	1456	2025	2591	3158	3725
τ_{OFF}	1226	1786	2347	2910	3473	4039
q_{c_1}	0.564	0.546	0.535	0.529	0.525	0.521
$q_{c_1}^*$	0.436	0.454	0.465	0.471	0.475	0.479
b	2.78	2.78	2.78	2.78	2.78	2.78

Section III-D. The respectively obtained channel lengths are provided in Table I; a sketch of the actual network is shown in Fig. 8, where each node has the structure illustrated in Fig. 2b. In fact, the generated network of Fig. 8 could be directly used in order to fabricate a physical realization of the ring network. Moreover, it can be clearly seen in Table I, that the channels indeed satisfy the conditions for dimensioning defined by Eqs. (7), (8), (12), and (15). Note that those dimensionings and, by this, the specification of the ring network has been determined fully automatically in less than a second, i.e., in negligible time.

B. Generating Droplet Sequences

Next, the droplet sequences realizing respective experiments (i.e., the distances between the payload and the header droplets

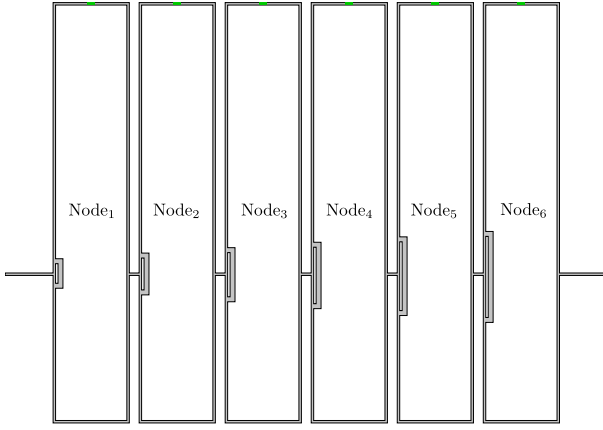


Fig. 8: Realization of the ring-network.

TABLE II: Resulting droplet distances (single-node approach).

	Modules					
	m_1	m_2	m_3	m_4	m_5	m_6
$d_{H,P}$ in μm	881	1341	1905	2469	3034	3599

which have to be employed for this purpose) are generated. Recall that, a single-node and a multi-node approach is available for this purpose. In the former approach, a single distance $d_{H,P}$ for each module/node m_i to be executed already exists. Using the methods described in Section III-C and Section III-D as well as assuming the design composed of six nodes as generated above, leads to the respective distances as shown in Table II, which were simply obtained by computing Eq. (16).

Following the multi-node approach, droplet sequences have to be determined for each (sub-)experiment individually. To this end, the procedure proposed in Section III-E and the conditions determined in Section III-C are applied. Fig. 7b again summarizes the correspondingly needed steps. Results for some selected representative experiments are provided in Table III. Here, each line first provides the (sub-)experiment followed by a line providing the correspondingly needed droplet distances at injection time (cf. Fig. 6b). All these results have, again, been generated in negligible time (a typically droplet sequence generation for the multi-node approach with 6 nodes requires approx. 3 CPU seconds on average).

C. Discussion

Overall, the design of a ring network and the droplet sequences generated for corresponding (sub-)experiments can be achieved by following the concepts and procedures from Section III and the steps summarized in Fig. 7a and Fig. 7b, respectively. In all our evaluations (including a large variety of considered experiments and corresponding droplet sequences), this led to the desired results in negligible time. However, there might be the (although unlikely) possibility that the proposed procedure (in particular, the one described at the end of Section III-E) does not converge. In this case, another approach would be to discretize the droplet distance between the payload droplet and the header droplet at the input channel of the network and, afterwards, test all available values with a

TABLE III: Resulting droplet distances (multi-node approach).

Droplet Distances in μm						
	$d_{H_1,P}$	d_{P,H_2}	d_{H_2,H_3}	d_{H_3,H_4}	d_{H_4,H_5}	d_{H_5,H_6}
$E_1 = m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_4 \rightarrow m_5 \rightarrow m_6$	881	41230	41867	41956	42047	42127
$E_2 = m_1 \rightarrow m_3 \rightarrow m_4 \rightarrow m_6$	881	40678	42185	41475		
$E_3 = m_2 \rightarrow m_4 \rightarrow m_5$	1341	40380	42264			
$E_4 = m_5 \rightarrow m_6$	3034	40195				
$E_5 = m_4 \rightarrow m_6$	2469	39890				
$E_6 = m_4 \rightarrow m_5 \rightarrow m_6$	2469	40446	42165			
$E_7 = m_3 \rightarrow m_6$	1905	39594				
$E_8 = m_3 \rightarrow m_5 \rightarrow m_6$	1905	40138	42378			
$E_9 = m_3 \rightarrow m_4$	1905	40700				
$E_{10} = m_3 \rightarrow m_4 \rightarrow m_5 \rightarrow m_6$	1905	40700	42064	42130		
$E_{11} = m_2 \rightarrow m_5 \rightarrow m_6$	1341	39828	42568			
$E_{12} = m_2 \rightarrow m_4$	1341	40380				
$E_{13} = m_2 \rightarrow m_4 \rightarrow m_5 \rightarrow m_6$	1341	40380	42264	42162		
$E_{14} = m_2 \rightarrow m_3$	1341	40933				
$E_{15} = m_2 \rightarrow m_3 \rightarrow m_5$	1341	40933	41381			
$E_{16} = m_2 \rightarrow m_3 \rightarrow m_5 \rightarrow m_6$	1341	40933	41381	42384		
$E_{17} = m_2 \rightarrow m_3 \rightarrow m_4 \rightarrow m_6$	1341	40933	41964	41464		
$E_{18} = m_2 \rightarrow m_3 \rightarrow m_4 \rightarrow m_5$	1341	40933	41964	42048		
$E_{19} = m_1 \rightarrow m_6$	881	38993				
$E_{20} = m_1 \rightarrow m_5$	881	39556				
$E_{21} = m_1 \rightarrow m_4 \rightarrow m_6$	881	40126	41926			
$E_{22} = m_1 \rightarrow m_4 \rightarrow m_5 \rightarrow m_6$	881	40126	42483	42169		
$E_{23} = m_1 \rightarrow m_3 \rightarrow m_6$	881	40678	41072			
$E_{24} = m_1 \rightarrow m_3 \rightarrow m_5$	881	40678	41615			
$E_{25} = m_1 \rightarrow m_3 \rightarrow m_4$	881	40678	42185			
$E_{26} = m_1 \rightarrow m_3 \rightarrow m_4 \rightarrow m_5 \rightarrow m_6$	881	40678	42185	42063	42122	
$E_{27} = m_1 \rightarrow m_2 \rightarrow m_6$	881	41230	40181			
$E_{28} = m_1 \rightarrow m_2 \rightarrow m_5$	881	41230	40750			
$E_{29} = m_1 \rightarrow m_2 \rightarrow m_5 \rightarrow m_6$	881	41230	40750	42568		
$E_{30} = m_1 \rightarrow m_2 \rightarrow m_4 \rightarrow m_6$	881	41230	41303	41700		
$E_{31} = m_1 \rightarrow m_2 \rightarrow m_4 \rightarrow m_5$	881	41230	41303	42278		
$E_{32} = m_1 \rightarrow m_2 \rightarrow m_4 \rightarrow m_5 \rightarrow m_6$	881	41230	41303	42278	42149	
$E_{33} = m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_5$	881	41230	41867	41375		
$E_{34} = m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_4$	881	41230	41867	41956		
$E_{35} = m_1 \rightarrow m_2 \rightarrow m_3 \rightarrow m_4 \rightarrow m_6$	881	41230	41867	41956	41463	

corresponding simulator (such as, e.g., provided in [13], [14], [15]). If no test value yields the required distance in front of the corresponding node (even under a discretization, which “perfectly” reflects the real world), the method shows that a corresponding solution is not physically possible. As a result, the designer has to check the proposed design conditions and has to consider an alternative configuration.

Moreover, once the design and the droplet sequences for a network with N nodes are available, the results can also be applied for all networks with less than N nodes. For example, when we assume that a network with 4 nodes is sufficient enough for our experiments, but we already have the results for the proposed network with 6 nodes, we can simply omit the dimensions of the 5th and 6th node inside Table I, in order to get a design with 4 nodes. The same method can be applied to get the droplet sequences for a desired (sub-)experiment. More precisely, for each occurrence of the 5th and 6th module inside a (sub-)experiment, the last droplet distance in Table III can be omitted.

Finally, all results obtained by the proposed approach have been validated using simulation engines proposed in [13], [14], [15]. They confirmed that the resulting design as well as the obtained droplet sequences indeed realize the desired behavior.

V. CONCLUSION

In this work, we proposed design automation methods for droplet-based microfluidic ring networks. Since designing such a ring network is a non-trivial problem, we first derived the respectively needed design conditions to be satisfied. They give designers the ability to realize an actual design of a ring network or directly generate them using the proposed automatic methods. Moreover, we discussed the addressing scheme which is needed in order to realize the desired experiments in a correct fashion. Also here, respective conditions to be satisfied as well as corresponding automatic methods are provided. Finally, the resulting design automation approaches have been demonstrated in a case study and validated using state-of-the-art simulators.

Future work considers the extension of these methods so that, additionally, fabrication effects are incorporated. This particularly includes the consideration of possible faults and/or inaccuracies which, thus far, frequently occur in today’s fabrication processes. The methods proposed in this work allow, in principle, to incorporate those effects by adjusting the dimensions of the affected channels or the corresponding factor used in Eq. (4). To this end, recent solutions can be exploited, which are able to analyze the robustness of a microfluidic device [24]. This would allow to design a ring-network in such a way, that effects of inaccuracies and/or defects are mitigated and, thus, ensures that the network works as intended, even when such defects occur during the fabrication process.

VI. ACKNOWLEDGMENT

This work has partially been supported by the LIT Secure and Correct Systems Lab funded by the State of Upper Austria.

REFERENCES

- [1] G. M. Whitesides, “The origins and the future of microfluidics,” *Nature*, vol. 442, no. 7101, pp. 368–373, 2006.
- [2] D. Mark, S. Haeberle, G. Roth, F. von Stetten, and R. Zengerle, “Microfluidic Lab-on-a-Chip platforms: requirements, characteristics and applications,” *Chemical Society Reviews*, vol. 39, no. 3, pp. 1153–1182, 2010.
- [3] P. S. Dittrich and A. Manz, “Lab-on-a-chip: microfluidics in drug discovery,” *Nature Reviews Drug Discovery*, vol. 5, no. 3, p. 210, 2006.
- [4] S.-Y. Teh, R. Lin, L.-H. Hung, and A. P. Lee, “Droplet microfluidics,” *Lab on a Chip*, vol. 8, pp. 198–220, 2008.
- [5] E. De Leo, L. Galluccio, A. Lombardo, and G. Morabito, “Networked labs-on-a-chip (NLoC): Introducing networking technologies in microfluidic systems,” *Nano Communication Networks*, vol. 3, no. 4, pp. 217–228, 2012.
- [6] 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.
- [7] A. Grimmer, W. Haselmayr, and R. Wille, “Automated dimensioning of Networked Labs-on-Chip,” *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2018.
- [8] —, “Automatic droplet sequence generation for microfluidic networks with passive droplet routing,” *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2018.
- [9] A. Biral and A. Zanella, “Introducing purely hydrodynamic networking functionalities into microfluidic systems,” *Nano Communication Networks*, vol. 4, no. 4, pp. 205–215, 2013.
- [10] A. Zanella and A. Biral, “Design and analysis of a microfluidic bus network with bypass channels,” in *Int’l Conf. on Communications*. IEEE, 2014, pp. 3993–3998.
- [11] 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.
- [12] W. Haselmayr, A. Biral, A. Grimmer, A. Zanella, A. Springer, and R. Wille, “Addressing multiple nodes in Networked Labs-on-Chips without payload re-injection,” in *Int’l Conf. on Communications*, 2017.
- [13] A. Biral, D. Zordan, and A. Zanella, “Modeling, simulation and experimentation of droplet-based microfluidic networks,” *Trans. on Molecular, Biological, and Multi-scale Communications*, vol. 1, no. 2, pp. 122–134, 2015.
- [14] A. Grimmer, M. Hamidović, W. Haselmayr, and R. Wille, “Advanced simulation of droplet microfluidics,” *Journal on Emerging Technologies in Computing Systems*, 2019.
- [15] A. Grimmer, X. Chen, M. Hamidović, W. Haselmayr, C. L. Ren, and R. Wille, “Simulation before fabrication: a case study on the utilization of simulators for the design of droplet microfluidic networks,” *RSC Advances*, vol. 8, pp. 34733–34742, 2018. [Online]. Available: <http://dx.doi.org/10.1039/C8RA05531A>
- [16] 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. 34104–34104, 2006.
- [17] G. Castorina, M. Reno, L. Galluccio, and A. Lombardo, “Microfluidic networking: Switching multidroplet frames to improve signaling overhead,” *Nano Communication Networks*, vol. 14, pp. 48–59, 2017.
- [18] M. Hamidović, W. Haselmayr, A. Grimmer, R. Wille, and A. Springer, “Passive droplet control in microfluidic networks: A survey and new perspectives on their practical realization,” *Nano Communication Networks*, vol. 19, pp. 33–46, 2019.
- [19] M. Hamidović, U. Marta, H. Bridle, D. Hamidović, G. Fink, R. Wille, A. Springer, and W. Haselmayr, “Off-chip-controlled droplet-on-demand method for precise sample handling,” *ACS Chemical Neuroscience*, 2020.
- [20] T. Glawdel and C. L. Ren, “Global network design for robust operation of microfluidic droplet generators with pressure-driven flow,” *Microfluidics and Nanofluidics*, vol. 13, no. 3, pp. 469–480, 2012.
- [21] 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.

- [22] H. Bruus, *Theoretical microfluidics*. Oxford university press Oxford, 2008, vol. 18.
- [23] K. W. Oh, K. Lee, B. Ahn, and E. P. Furlani, "Design of pressure-driven microfluidic networks using electric circuit analogy," *Lab on a Chip*, vol. 12, no. 3, pp. 515–545, 2012.
- [24] G. Fink, A. G. M. Hamidović, W. Haselmayr, and R. Wille, "Robustness analysis for droplet-based microfluidic networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.



Robert Wille (M'06–SM'15) is Full Professor at the Johannes Kepler University Linz, Austria, and Chief Scientific Officer at the Software Competence Center Hagenberg, Austria. He received the Diploma and Dr.-Ing. degrees in Computer Science from the University of Bremen, Germany, in 2006 and 2009, respectively. Since then, he worked at the University of Bremen, the German Research Center for Artificial Intelligence (DFKI), the University of Applied Science of Bremen, the University of Potsdam, and the Technical University Dresden. Since 2015, he is working in Linz/Hagenberg. His research interests are in the design of circuits and systems for both conventional and emerging technologies. In these areas, he published more than 300 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. For his research, he was awarded, e.g., with a Best Paper Award at ICCAD, a DAC Under-40 Innovator Award, a Google Research Award, and more.



Gerold Fink received the Master's degree in mechatronics from the Johannes Kepler University Linz, Austria, in 2019. Currently, he is a Ph.D. student at the Institute of Integrated Circuits at the Johannes Kepler University. His research area focuses on simulations and design automations for microfluidic networks.



Medina Hamidović has received her B.Sc. degree in electrical engineering at the University of Tuzla in 2014. In 2017, Hamidović received her joint Master's degree in electrical engineering from Heriot-Watt University (UK), University of South-East Norway and Budapest University of Technology and Economics. At the moment, Hamidović is a Ph.D. researcher at the Institute for Communications Engineering and RF-Systems at the Johannes Kepler University Linz (Austria). Her research is focused on the area of molecular communications and microfluidic networks.

idic networks.



Werner Haselmayr (S'08–M'13) is an Assistant Professor at the Institute for Communications Engineering and RF-Systems, Johannes Kepler University Linz, Austria. He received the Ph.D. degree in mechatronics from the same university in 2013. His research interests include the design and analysis of synthetic molecular communication systems and communications and networking in droplet-based microfluidic systems. He has given several invited talks and tutorials on various aspects of droplet-based communications and networking. He has authored 2 book chapters and more than 60 papers, appeared in top-level international peer-reviewed journals and conference proceedings. Currently, he serves as Associate Editor for the IEEE Transactions on Molecular, Biological, and Multi-Scale Communications.

thored 2 book chapters and more than 60 papers, appeared in top-level international peer-reviewed journals and conference proceedings. Currently, he serves as Associate Editor for the IEEE Transactions on Molecular, Biological, and Multi-Scale Communications.