

# Performance and Cost Analysis of NoC-Inspired Virtual Topologies for Digital Microfluidic Biochips

Daniel Grissom

Department of Computer Science  
Azusa Pacific University  
Azusa, CA, USA  
dgrissom@apu.edu

Philip Brisk

Department of Computer Science and Engineering  
University of California, Riverside  
Riverside, CA, USA  
philip@cs.ucr.edu

**Abstract**—Virtual topologies simplify the process of compiling assays to execute on digital microfluidic biochips (DMFBs). This paper evaluates the performance and cost of a virtual topology inspired by networks-on-chip (NoCs). The throughput of several deadlock-free droplet routing protocols is compared on synthetic traffic patterns that are widely used to evaluate semiconductor NoCs. The cost is the number of control pins required for actuation and the number of PCB layers required to route the chip; by eliminating unused pins, the virtual topology is cheaper than a direct-addressing DMFB, especially as chip size increases.

**Index Terms**—Digital Microfluidic Biochip (DMFB), Virtual Topology, Network-on-Chip (NoC), Droplet Routing, Deadlock

## I. INTRODUCTION

A digital microfluidic biochip (DMFB), shown in Fig. 1(a), manipulates discrete droplets of liquid on a two-dimensional electrode grid. As an alternative to traditional benchtop chemistry, DMFBs offer miniaturization, automation, and software programmability. Cyber-physical DMFBs now integrate a variety of electrical and electrochemical sensors. Efficient interpreters and just-in-time compilers are necessary to enable execution of assays (biochemical protocols) that process sensory feedback and make control-flow decisions in real-time. Virtual topologies [4-8], which segregate a DMFB's electrodes in dedicated regions for droplet transport and other operations, provide a clean and efficient abstraction layer that drastically simplifies the compilation process.

This paper evaluates the performance and cost of a virtual topology inspired by two-dimensional mesh networks-on-chip (NoCs). We measure the throughput of several deadlock-free droplet routing protocols on the virtual topology using a variety of synthetic traffic patterns that are widely used to evaluate semiconductor NoC performance. We then compute the number of control pins and PCB layers required to realize a DMFB using this virtual topology, and compare the result to a direct-addressing DMFB; our results demonstrate significant reduction in the number of control pins and PCB layers.

Fig. 1(b) depicts a cross-sectional view of a DMFB: a droplet is centered on top of electrode CE2 and overlaps two adjacent electrodes, CE1 and CE3. Applying a voltage to CE2, but not CE1 or CE3 holds the droplet in-place; deactivating CE2 and activating CE1 or CE3 moves the droplet left or right, respectively; deactivating CE2 and activating CE1 and CE3 simultaneously splits the droplet in two.

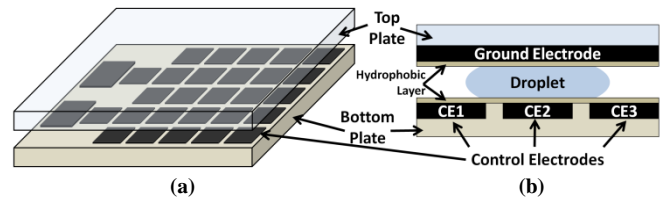


Fig. 1. (a) A DMFB consists of a 2D array of electrodes; (b) a DMFB cross-sectional showing that a droplet overlaps neighboring electrodes.

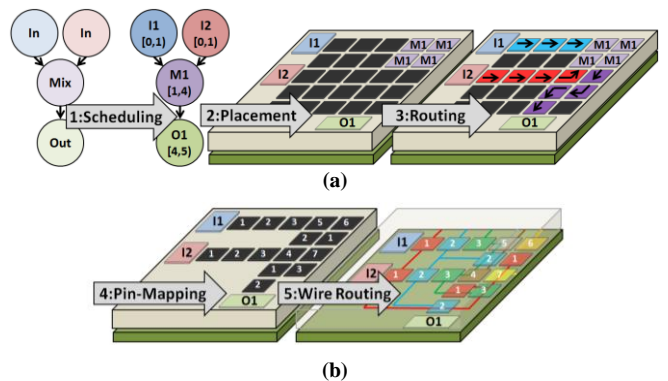


Fig. 2. A DMFB compiler performs three fluidic synthesis steps (scheduling, placement and routing); (b) two additional steps synthesize the control and PCB hardware (pin-mapping and wire-routing).

The instruction set of a DMFB includes droplet transport, splitting, merging, mixing, and storage; integration of sensors and/or actuators (e.g., heating elements) extends the instruction set with new capabilities: to use an extension, a droplet must be routed to an appropriate location on the chip and held in place during sensing or actuation. DMFBs offer abundant spatial parallelism and can perform many concurrent operations.

As shown in Fig. 2, a DMFB compiler must solve three interdependent NP-complete problems that determine fluidic movement: (1) scheduling of assay operations; (2) spatial placement of operations on the DMFB surface; and (3) routing droplets between placed operations. Two additional stages, pin mapping and wire routing, assist with the automatic layout of a printed circuit board (PCB) on which the DMFB resides. Pin-mapping facilitates control pin sharing, which reduces the number of signals that must be delivered to control the DMFB. Wire routing connects each control input pin to the electrode(s) that it drives; for large chips, multiple PCB layers may be needed, each of which increases the overall cost [11].

## II. RELATED WORK: VIRTUAL TOPOLOGIES

A cell is the abstraction of a square region on top of each electrode in a DMFB. Contiguous groups of one or more cells can perform all of the basic DMFB operations described in the preceding section. A virtual topology defines distinct regions of the chip for droplet routing (e.g., a network of streets) that transport droplets to dedicated modules that perform other operations (mixing, storage, detection, etc.).

Griffith and Akella proposed the first virtual topology [4]; although it separated a droplet transport network from modules that perform operations, it could not guarantee deadlock-free droplet routing; to prevent deadlocks, Griffith and Akella limit the number of droplets that can be injected into the system.

Fig. 3 shows a virtual topology introduced by Grissom and Brisk [7]; by limiting the number of droplets that any module can store to at most two, and dedicating specific I/O locations for each of two droplets that may enter a module, deadlock-free routing using this virtual topology is achieved. (More droplets can be safely stored by increasing the module dimensions, e.g., to 5x5). The drawback is that it is imposed on top of a direct-addressing DMFB, which is expensive in terms of the number of PCB layers obtained after wire routing.

Fig. 4 shows a low-cost virtual topology introduced by Grissom and Brisk [5]. It employs a pin-constrained design, and can be routed in two PCB layers (one layer if the vertical busses on the left and right are removed); thus, it is very low-cost; however, it lacks the capacity to transport many droplets concurrently, and thus has sub-optimal performance.

Fig. 5 illustrates the virtual topology that is the subject of this paper [6, 8]. It is inspired by two-dimensional mesh-based NoCs for multi-core semiconductor chips. The cells shown in white are not used so no electrodes are needed there; this reduces the number of control pins and the number of wires that need to be routed to drive electrodes; our experiments demonstrate that this reduces the cost of the chip significantly. This topology is compatible with various deadlock-free droplet routing protocols for mesh networks [1-3], therefore it offers exceptional throughput with low computational overhead. Furthermore, each module can store up to four droplets, rather than two. Altogether, this virtual topology offers a favorable cost-performance tradeoff, especially for assays that require high throughput transport of a large number of droplets.

## III. DROPLET ROUTING WITH A VIRTUAL TOPOLOGY

The virtual topology in Fig. 5 is based on two-dimensional NoCs. In a NoC, the network is composed of routers (nodes); each node transmits/receives packets to/from its neighbors to the north, south, east, and west. When a packet arrives, the node queues it in an input buffer; when the node examines a packet, it is delivered to the local processor or transmitted to an adjacent node, chosen by the routing protocol. Deadlock occurs when nodes in a cycle are waiting for one another. In a two-dimensional mesh, packets in-flight can make eight possible turns, potentially forming cycles in two different directions, as shown in Fig. 6. As shown in Fig. 7, deadlock-free routing in a mesh network is achieved by prohibiting certain turns [2, 3] or by restricting the locations where certain turns may occur [1].

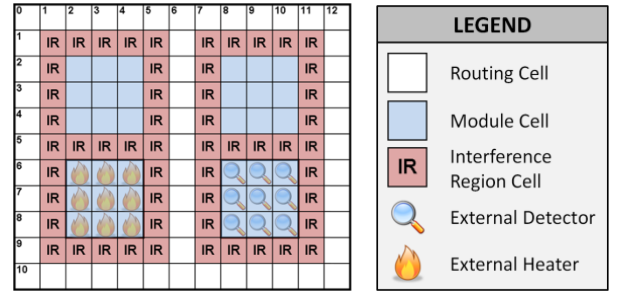


Fig. 3. A high-cost, medium-throughput virtual topology that is imposed on a direct-addressing DMFB [7].

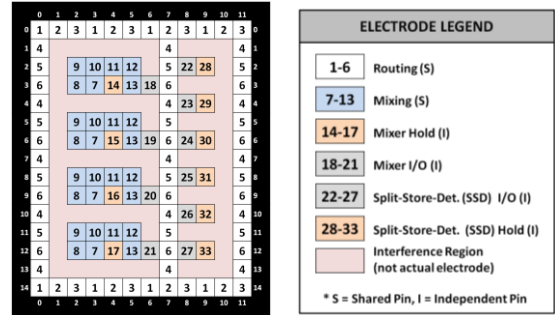


Fig. 4. A low-cost, low-throughput pin-constrained virtual topology [5].

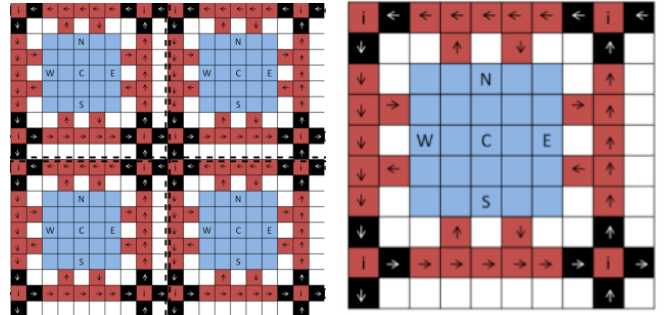


Fig. 5. A medium-cost, high-throughput pin-constrained virtual topology [6, 8]. In lieu of a legend, a single tile is shown on the right. It includes a work module (blue) and droplet transport network (red/black); white cells are unaddressed and do not have electrodes.

Dimension-order routing (DOR) [2] (XY/YX; Fig. 7(b) and (c)) forces each packet to travel along one dimension and then the other, making exactly one turn along the way; this limits adaptability to localized congestion. Three routing protocols based on the turn model [3] (Negative-First, North-Last, West-First; Fig. 7(d), (e), and (f)) prohibit one turn from each cycle; they prevent deadlock with better adaptability than DOR. If a packet can be routed in more than one direction upon reaching a node, it is routed to the lowest-cost neighbor, based on network congestion analysis. The odd-even turn model [1] (Fig. 7(g) and (h)) prevents deadlock by prohibiting certain turns in certain columns, allowing greater flexibility than DOR or the more restrictive turn model-based protocols.

The interested reader may refer to [6] for more details about how we adapt these deadlock-free routing algorithms to DMFBs using the virtual topology shown in Fig. 5.

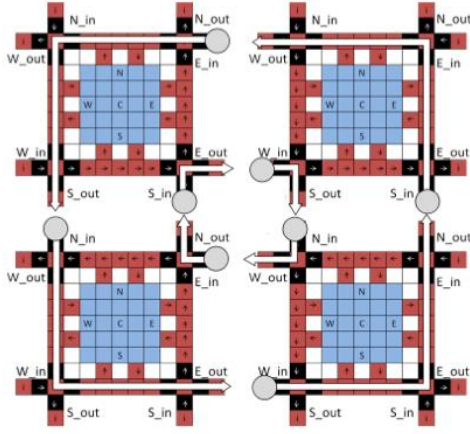


Fig. 6. Clockwise and counter-clockwise turns that may occur when routing droplets using the virtual topology in Fig. 5

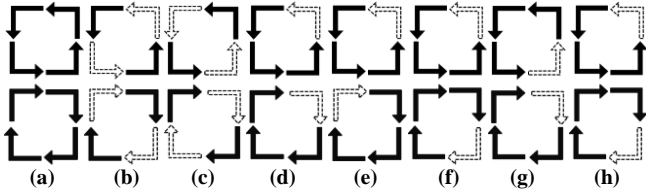


Fig. 7. (a) All possible turns in a 2D-mesh network; (b) XY turns; (c) YX turns; (d) Negative-First turns; (e) North-Last turns; (f) West-First turns; (g) Turns prevented in odd columns; (h) Turns prevented in even columns. NOTE: Solid/dashed arrows represent permitted/forbidden turns.

Let  $T$  be the current tile,  $D$  be the droplet being routed,  $d$  the direction (north, south, east, west) considered, and  $T_d$  be the neighboring tile in direction  $d$  from tile  $T$ . Let  $P_1$  and  $P_2$  be constant penalties, and  $Dir(T_d, D) = 1$  if routing from tile  $T$  to  $T_d$  moves  $D$  closer to its destination on any dimension, and  $0$  otherwise. Let  $Drops(T_d)$  be the number of droplets in tile  $T_d$ .

The cost to route droplet  $D$  from tile  $T$  in direction  $d$  is  $Cost(T, d, D) = P_1 \times Dir(T_d, D) + P_2 \times Drops(T_d)$ ; the droplet is routed in the direction that minimizes the cost function.

#### IV. SIMULATION RESULTS

##### A. Droplet Routing

We created an event-driven, DMFB simulator to evaluate the throughput of droplet routing protocols using the virtual topology in Fig. 5. We used four established network traffic patterns (random, neighbor, bit-complement and transpose) taken from NoC routing studies [10]. These patterns were used to populate the system and schedule droplet destinations. Simulations were performed on a 4x4 and 8x8 mesh of tiles. All border tiles have a fluid I/O port on each exposed side; the 4x4 and 8x8 arrays have 32 and 64 I/O ports respectively.

For the first 100 simulated cycles, droplets are generated at each input reservoir at a user-specified injection rate. For example, if 4x4 mesh is coupled with a 75% injection rate, then  $(100 \text{ cycles} \times 0.75 \text{ droplets/cycle} \times 16 \text{ input ports}) = 1,200$  droplets will be injected into the system. We generate a schedule specifying two intermediate modules and one output port for each droplet, based on the chosen traffic pattern. The schedule is applied to all six routing protocols for each test.

We vary the injection rate of droplets into the system and perform 100 runs for each traffic pattern and routing protocol at each injection rate. For each traffic pattern, we report the average throughput for each routing protocol as a function of injection rate. Throughput in this context is the simulated number of droplets per cycle that complete their routes; higher throughput implies greater routing concurrency.

Figs. 8 and 9 report results for the 4x4 and 8x8 arrays, respectively. At the lower end of the spectrum, increasing the injection rate increased the throughput of the system, as more droplets per cycle complete. Inevitably, a saturation point is eventually reached; at this point, injecting more droplets into the system leads to congestion, and throughput plateaued as a result: to prevent deadlocks, the protocols must limit the number of droplets injected into the routing network when congestion is high; please consult [6] for details.

Each chart in Figs. 8 and 9 ranks the routing protocols from top-to-bottom, where the topmost algorithm yields the highest throughput at an injection rate of 20%. Looking at the results in aggregation, the DOR routers (XY/YX) tend to yield the highest throughput. The reason is that adaptive routing suffers a high penalty, because the decision to direct traffic away from a direct route to a destination leads to a very long detour (when counting cells). Often, the cost of waiting a few extra cycles for the congestion to alleviate itself is less than the cost of routing away from congestion, which includes a long route around the perimeter of one (or more) tile(s). Although adaptive routing can still offer some benefits over DOR, it is quite clear that they must be much more conservative in making routing decisions than their semiconductor NoC counterparts.

##### B. PCB Routing Results

Fig. 10 compares the number of control pins (addressed electrodes and PCB layers) for a 20x20 direct-addressing DMFB and a 4x4 virtual topology imposed on a 20x20 DMFB. The direct-addressing DMFB addresses 400 electrodes and requires three PCB layers; the virtual topology addresses 276 electrodes (a 31% reduction) and requires two PCB layers. (The number of PCB layers was determined by computing an escape route for each electrode, as described in [9]). Thus, the virtual topology is cheaper when all PCB costs are taken into account. In its favor, the direct-addressing DMFB can support more concurrent operations than the virtual topology, however, these benefits come at a non-negligible cost.

Table I reports that the virtual topology achieves a much lower cost than a direct-addressing DMFB of equal dimensions. The cost of the direct-addressing DMFBs increases at a greater rate than that of the virtual topology as the chip size increases: at 20x20, the virtual topology required one less PCB layer than the direct-addressing chip; at 40x40 and 50x50, it requires three fewer layers.

It is important to note that the actual PCB cost includes more than the number of control pins and the number of layers. A typical PCB should include space for a microcontroller (which provides an interface between a PC that controls the DMFB and the device itself) and shift registers to hold signals generated by the microcontroller (if the number of addressed electrodes exceeds the number of microcontroller outputs).

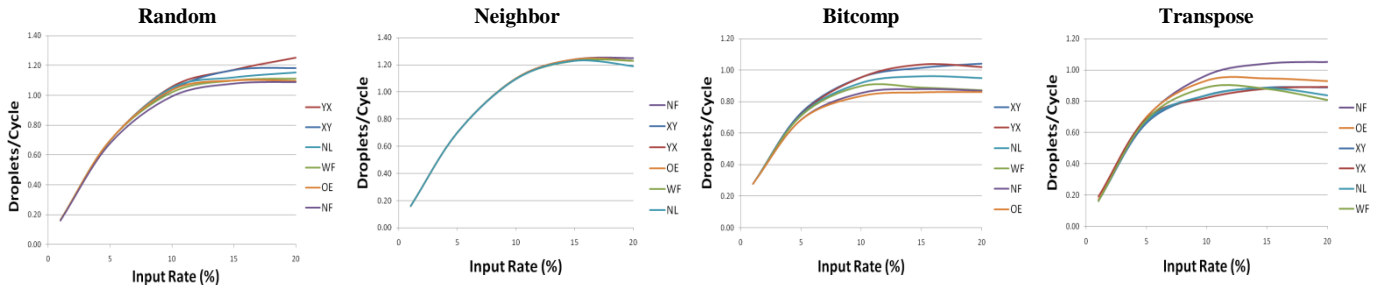


Fig. 8. Throughput as a function of injection rate for four synthetic traffic patterns on a 4x4 array of tiles. The list of routing protocols on the right of each chart orders them from highest (top) to lowest (bottom) in terms of throughput.

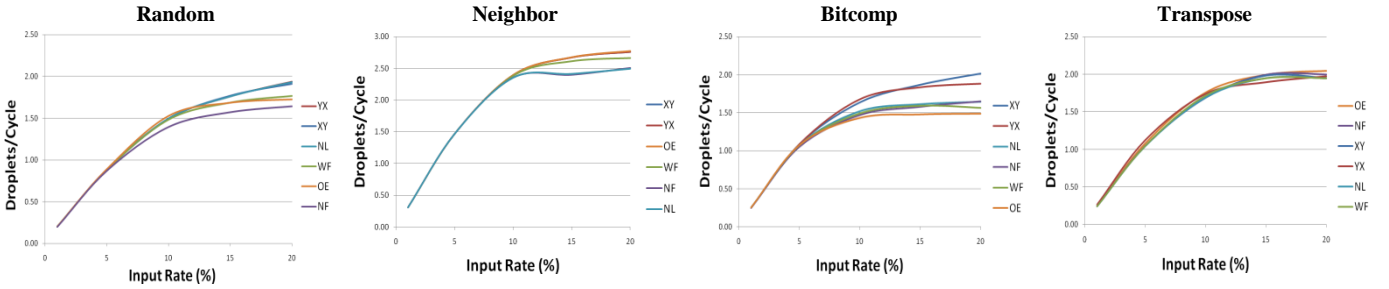


Fig. 9. Throughput as a function of injection rate for four synthetic traffic patterns on an 8x8 array of tiles. The list of routing protocols on the right of each chart orders them from highest (top) to lowest (bottom) in terms of throughput.

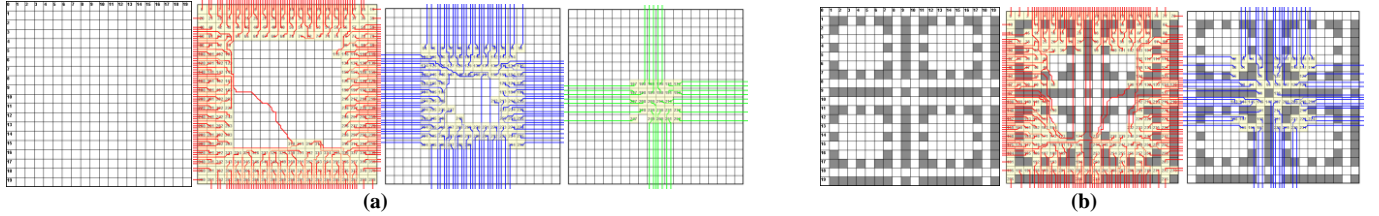


Fig. 10. (a) A 20x20 direct-addressing DMFB with 400 addressed electrodes requires three PCB layers. (b) A 4x4 virtual topology imposed on a 20x20 DMFB with 276 addressed electrodes (gray electrodes are unaddressed) requires two PCB layers. PCB layers #1, 2, and 3 are shown in red, blue, and green respectively.

TABLE I. THE NUMBER OF CONTROL PINS AND PCB LAYERS FOR DMFBs EMPLOYING DIRECT ADDRESSING OR OUR VIRTUAL TOPOLOGY (FIG. 5).

Direct Addressing (DA) vs. Virtual Topology (VT)								
DMFB Size/Type	20x20		30x30		40x40		50x50	
	DA	VT	DA	VT	DA	VT	DA	VT
Pin count	400	276	900	621	1600	1104	2500	1725
PCB Layers	3	2	5	3	7	4	9	6

## V. CONCLUSION

We have demonstrated that non-adaptive, deadlock-free routing algorithms outperform adaptive routing algorithms for a NoC-based virtual topology imposed on a DMFB. We also showed that removing electrodes unaddressed by the virtual topology reduces the pin count and number of PCB layers compared to a direct addressing DMFBs of equal dimensions.

## ACKNOWLEDGMENT

This work was supported in part by NSF Grant CNS-1035603. Daniel Grissom was supported by an NSF Graduate Research Fellowship and a UC Riverside Dissertation Year Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NSF.

## REFERENCES

- [1] G. Chiu, "The odd-even turn model for adaptive routing," IEEE TPDS, 11(7): 1001-1017, July 2010.
- [2] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," IEEE Trans. Computers, C-36(5): 547-553, May 1987.
- [3] C. Glass and L. Ni, "The turn model for adaptive routing," J. ACM, 41(5): 874-902, September 1994.
- [4] E. J. Griffith, S. Akella, and M. K. Goldberg, "Performance characterization of a reconfigurable planar-array digital microfluidic system," IEEE TCAD 25(2): 340-352, February 2006.
- [5] D. Grissom and P. Brisk, "A field-programmable pin-constrained digital microfluidic biochip," DAC, June 2013, article #46.
- [6] D. Grissom and P. Brisk, "A high-performance online assay interpreter for digital microfluidic biochips," GLS-VLSI, May 2012, pp. 103-106.
- [7] D. Grissom and P. Brisk, "Fast online synthesis of digital microfluidic biochips," IEEE TCAD, 33(3): 356-359, March 2014.
- [8] D. Grissom, C. Curtis, and P. Brisk, "Interpreting assays with control flow on digital microfluidic biochips," ACM JETC 10(3): article #24, April 2014.
- [9] J. McDaniel, D. Grissom, and P. Brisk, "Multi-terminal PCB escape routing for digital microfluidic biochips using negotiated congestion," VLSI-SoC, October, 2014.
- [10] Y. Pan et al., "Firefly: illuminating future network-on-chip with nanophotonics," ISCA, June 2009, pp. 429-440.
- [11] T-W. Huang, S-Y. Yeh, and T-Y. Ho, "A network-flow based pin-count aware routing algorithm for broadcast-addressing EWOD chips," IEEE TCAD 30(12): 1786-1799, Dec., 2011.