The Cost of Scaling a Reliable Interconnection Topology

Rachid Guerraoui and Alexandre Maurer

Abstract—In distributed computing, many papers try to evaluate the message complexity of a distributed system as a function of the number of nodes n. But what about the cost of *building* the distributed system itself? Assuming that we want to reliably connect n nodes, how does the total number of nodes of the network evolve with n? Addressing such a question lies at the heart of achieving scalability in cloud computing. In this paper, we give the explicit description of a distributed system of which any two of the n nodes, for any n, *remain connected* (by a path of alive nodes and channels) with probability at least μ, despite the very fact that (a) every other node or channel has an independent probability λ of failing, and (b) the number of channels connected to every node is physically bounded by a constant. We show however that if we also require any two of the n nodes to maintain a *balanced message throughput* with a constant probability, then $O(n\log^{1+\epsilon}n)$ additional intermediary nodes are sufficient, where $\epsilon > 0$ is an arbitrarily small constant.

Index Terms—Scalability, reliability, degree, throughput, network

1 Introduction

The growth of modern networks seems to be exceeding Moore's Law [2]. More and more computers are getting connected in cloud computing centers handling massive data storage [6], [8]. We talk for example about 60,000 cores for the Blue Brain Project [5] and over 100,000 for the CERN data center [1]. Companies like Google and Microsoft have data centers with millions of servers [3]. Not surprisingly, the problem of how to achieve *scalability* and effectively connect a very large number of computers has been extensively studied (see Related Works, Section 9). In particular, a lot of attention has been devoted to maintaining a reasonable message *throughput* (i.e., avoid traffic congestion), even when the size of the network increases. A major difficulty that hinders such scalability is the *bounded* (by a physical constant) capacity of network components (computers and channels): there is a maximal number of messages per second that a channel can transmit, and a maximal number of channels that a node (computer) can connect. A closer look at existing cloud constructions reveals in fact that, strictly speaking, traffic congestion increases when the size of the network increases. This is without even accounting for *failures*: when the size of the network increases, the probability that several components of the network *fail* also increases, making it even more difficult to maintain any stable throughput.

This paper asks the question of the theoretical price of scalability. Assume that we want to reliably connect n nodes while preserving a stable message throughput and a bounded degree. The *cost* of such a network is the total number of nodes required to build it (including the n nodes we want to connect). We seek to determine how this cost evolves with n.[1]

We consider the case of random failures. A natural approach is to consider the probability that the subgraph of correct nodes remains connected. However, when a graph with a bounded degree Δ becomes very large, this probability approaches zero. Indeed, as a node has at most Δ neighbors, it has a fixed probability to be cut off from the rest of the network. Over a large number of nodes, the probability to have at least one node in this situation approaches zero. We thus consider a more relaxed criteria: the minimal probability that two nodes (which can be any nodes) remain connected. As we show in this paper, it is actually possible to make this probability arbitrarily high, regardless of the failure rate.

While other solutions (see the Related Works section) can also tolerate random failures, their guarantees collapse when the network reaches a certain size (i.e., the goal probability approaches zero). The main goal of this paper is to fix this problem, together with additional constraints (such as preserving a constant throughput). The choice to focus on random failures is motivated by the observation that, in practice, most failures happen randomly (and are not selected by a centralized malicious agent). This paper provides several upper and lower bounds for the randomized model, but many problems remain open within this model. Note that, despite random failures, the problem is deterministic and requires a deterministic solution. Thus, random graphs cannot be a solution here. The hypercube graph cannot be a solution either, as it does not have a bounded degree (the degree increases with the number of nodes). We discuss the case of expander graphs in the Related Works section.

• *The authors are with the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne 1015, Switzerland.*
E-mail: {rachid.guerraoui, alexandre.maurer}@epfl.ch.

Manuscript received 4 Sept. 2017; revised 20 Apr. 2018; accepted 4 June 2018. Date of publication 8 June 2018; date of current version 1 Sept. 2020. (Corresponding author: Alexandre Maurer.)
Digital Object Identifier no. 10.1109/TDSC.2018.2845402

1. Note that we do not require the graph to grow gracefully with n here: the graph connecting $n+1$ nodes can be very different from the graph connecting n nodes.

In the following paragraphs, we informally explain the problems and their solutions. Formal definitions of the problems are provided in Section 3.

We proceed incrementally.

(1) We first address what we call the RBD (**R**eliable **B**ounded **D**egree) problem, on how to connect a set of nodes so that every pair can communicate (i.e., are connected by a path of alive nodes and channels) with probability at least μ, assuming that any other node or channel has an independent probability at most λ to crash [19], [30]. (We leave aside any throughput requirement as well as Byzantine failures in this first step.) Building a complete graph, connecting any two nodes with a channel is not a solution as the node degree (i.e., the number of channels connected to a given node) keeps increasing. In fact, the RBD problem might actually seem impossible without additional *intermediary* nodes between the n nodes (acting as routers and not necessarily reliably connected to the rest). When n increases, the diameter of the graph also increases: pairs of nodes become more distant from each other, inevitably dragging down the communication probability. Compensating for this loss of reliability by adding redundant paths between any pair of (distant) nodes is infeasible for the number of parallel paths is bounded by the maximal degree whereas the network diameter keeps increasing with n.

We show in this paper how to address the RBD problem (with no additional intermediate nodes). For any number of nodes n, we show how to build a graph of n nodes that ensures arbitrarily high reliability while preserving a bounded degree. We proceed in two substeps. We first solve the **W**eak *RBD* (WRBD) problem, whose goal is to reliably connect n nodes with a graph of bounded degree, by allowing to add intermediary nodes between these n nodes, provided that their number is $O(n)$ (at most linear in n). We do so by defining a recursive graph that ensures a constant communication probability between any two given nodes (independently of their distance) with a bounded degree, expressing the communication probability as a *convergent sequence*, and then a *tree-like* layered graph reliably connecting n nodes. We then use the solution to the WRBD problem to solve our seemingly stronger RBD problem, i.e., reliably connecting n nodes *without* intermediary nodes (the construction works with any graph solving the WRBD problem). The idea is to combine several instances of a WRBD graph, each instance reliably connecting a smaller number of nodes, and to make their intermediary nodes disappear by merging them with other nodes.

(2) We then address the problem of *message throughput*. We model the exchanges of messages by continuous *flows* of messages. Each of the n nodes needs to transmit the same flow of messages to the $n-1$ other nodes.[2] Assuming a bound, independent from n, on (1) the maximal *degree* of the network and (2) the maximal *flow* of the network, i.e., the maximal flow of messages crossing each node and channel, we address the *BDF* (**B**ounded **D**egree and **F**low) problem (first leaving aside the reliability requirement), which consists in finding a graph that enables to maintain the flow of messages between the n nodes. Again, the constraint on the degree prevents a complete graph directly connecting each pair of n nodes. Thus, some flows of messages will have to go through *intermediary* nodes (acting as routers). At first glance, one might consider using these intermediary nodes in a tree topology, of which the leaves would be the n nodes. However, a tree network is problematic for all messages would need to cross the root node, making the maximal flow increase with n. In fact, we prove that solving the BDF problem requires at least $\Omega(n \log n)$ intermediary nodes. Basically, the bounded degree implies a distance $\Omega(\log n)$ between most pairs of nodes, and the resulting amount of messages has to be distributed over a minimal number of intermediary nodes, due to the bounded capacity. We then describe a graph solving the BDF problem using $O(n \log n)$ intermediary nodes, which matches the lower bound. Essentially, our solution is again multi-layered, and consists in stacking $O(\log n)$ layers of $O(n)$ nodes each, and then crossing the flow of messages between each layer so that (1) the flow of messages crossing each node remains constant and (2) the flows of messages are uniformly mixed when reaching the last layer. We merge the first and the last layer of the graph, enabling each one of the n nodes to exchange messages with the $n-1$ other nodes.

(3) Finally, we combine the RBD and BDF problems and define the *RBDF* (**R**eliable **B**ounded **D**egree and **F**low) problem. As for RDB, we assume that each node and channel has a given probability λ to crash, and that each pair of nodes (among the n initial nodes) must keep exchanging the same flow of messages with probability μ. We also define a recursive graph that ensures reliable communication between any two nodes, at whatever distance they may be (w.r.t the parameters λ and μ). Then, we make a layer-by-layer product of this graph with the BDF multi-layered graph, in order to combine this reliability property with the bounded degree and flow properties. The number of intermediary nodes of the resulting graph then goes from $O(n \log n)$ to $O(n \log^{1+\epsilon} n)$, where ϵ is a positive constant that can be as small as wanted. In other words, the additional cost of the reliability property lies in a factor $\log^{\epsilon} n$, where ϵ can be as small as wanted.

Interestingly, all our constructions have an optimal (logarithmic) diameter. Besides, they can all be extended to tolerate Byzantine failures (when the failed components, i.e., nodes or channels, behaves arbitrarily), assuming the failure rate λ to be strictly smaller than 0.5, by (1) increasing the level of redundancy (compared to the case of crash failures) and (2) adding several layers of majority votes to eliminate malicious messages.

Overall, the goal of this paper is to show that a network can scale without limitation while tolerating constraints such as random failures, bounded degree, bounded throughput, or all together. All these constraints derive from intrinsic limitations of network components: a component is not perfectly reliable (random failures); a node cannot be plugged to an unbounded number of channels (bounded degree); a channel cannot hold an unbounded throughput (bounded throughput). In other words, we show that *bounded* characteristics of network components do not prevent the network from being *unbounded*.

The problems, while simple, require complex and nontrivial graph constructions to be solved. We do not claim

2. Here, "identical" means that any node p sends the same quantity of messages to any two nodes q and r, which does not mean that the messages sent to q and r are the same.

our solutions to these problems to be unique. In the related works section, we discuss alternative ways to solve these problems.

The Rest of the Paper is Organized as Follows. Section 2 presents our model and Section 3 defines the problems we address (WRBD, RBD, BDF, RBDF). Sections 4, 5, 6, and 7 present solutions to these problems and argue for their correctness. In Section 8, we prove our results in terms of cost. We present the related works in Section 9 and conclude in Section 10. In the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TDSC.2018.2845402, we discuss the logarithmic diameter of our solutions, and explain how our solutions can be generalized to handle Byzantine failures.

Due to space limitations, all proofs are delegated to the supplemental material, available online.

2 Model

A graph is a tuple $G = (V, E)$ where V is the set of *nodes* and E is the set of *channels*, modeled as a set with repetition of pairs of nodes $\{p, q\} \subseteq V$ (we enable multiple channels between p and q). The *degree* $\delta(v)$ of a node v is the number of channels (p, q) such that $p = v$ or $q = v$ (the number of channels connected to v). The *maximal degree* of graph G is $\max_{v \in V} \delta(v)$. A *path* connecting two nodes p and q is a sequence of nodes $(u_1, \ldots, u_m)$ such that $u_1 = p$, $u_m = q$ and $\forall i \in \{1, \ldots, m-1\}$, u_i and u_{i+1} are neighbors.

A *component* of a graph G is any node or channel of G. Each component of G can be either *correct* (functional) or *crashed* (failed). A *correct path* is a sequence of nodes $(p_1, \ldots, p_m)$ such that, $\forall i \in \{1, \ldots, m\}$, p_i is correct, and $\forall i \in \{1, \ldots, m-1\}$, there exists a correct channel $\{p_i, p_{i+1}\}$. Two nodes p and q are *connected* if there exists a correct path $(p_1, \ldots, p_m)$ such that $p_1 = p$ and $p_m = q$. We denote by $\lambda \in]0, 1[$ and $\mu \in]0, 1[$ two arbitrary constants.

Fluid Message Flow (FMF). Let $S \subseteq V$ be any arbitrary set of n nodes, with $n \geq 2$, representing the computers of the network that need to issue and exchange messages. The rest of the nodes are *intermediary* nodes corresponding to routers that forward the messages sent by the n computers of S: they do not issue messages of their own.

We consider a perfectly balanced distributed (peer-to-peer) system: each of the nodes of S sends the same quantity of messages to every other node. More precisely, we assume that each node $p \in S$ sends a flow of messages F, equally distributed between the $n-1$ other nodes of S.[3] Thus, for any two nodes p and q of S, p sends a flow of messages $F/(n-1)$ directed towards q. We now define the paths taken by these messages.

A *weighted path* is a tuple (P, α), where P is a path and α is an arbitrary coefficient. A weighted path represents a continuous flow of messages between two nodes p and q, where P is the path used by the messages, and α is the fraction of messages directed towards q. For any two nodes p and q of S, the flow of messages from p to q uses a set of weighted paths $R(p, q) = \{(P_1, \alpha_1), (P_2, \alpha_2), \ldots, (P_m, \alpha_m)\}$. The paths $P_1, P_2, \ldots, P_m$ are connecting p to q, and $\alpha_1 + \alpha_2 + + \alpha_m = 1$. For each path P_i, the coefficient α_i corresponds to the fraction of the flow of messages using the path P_i. We illustrate this structure through a simple example in Fig. 1.

Thus, path P_i receives a flow $\alpha_i F/(n-1)$ of messages from p to q. We call the function R the *routing map* of S

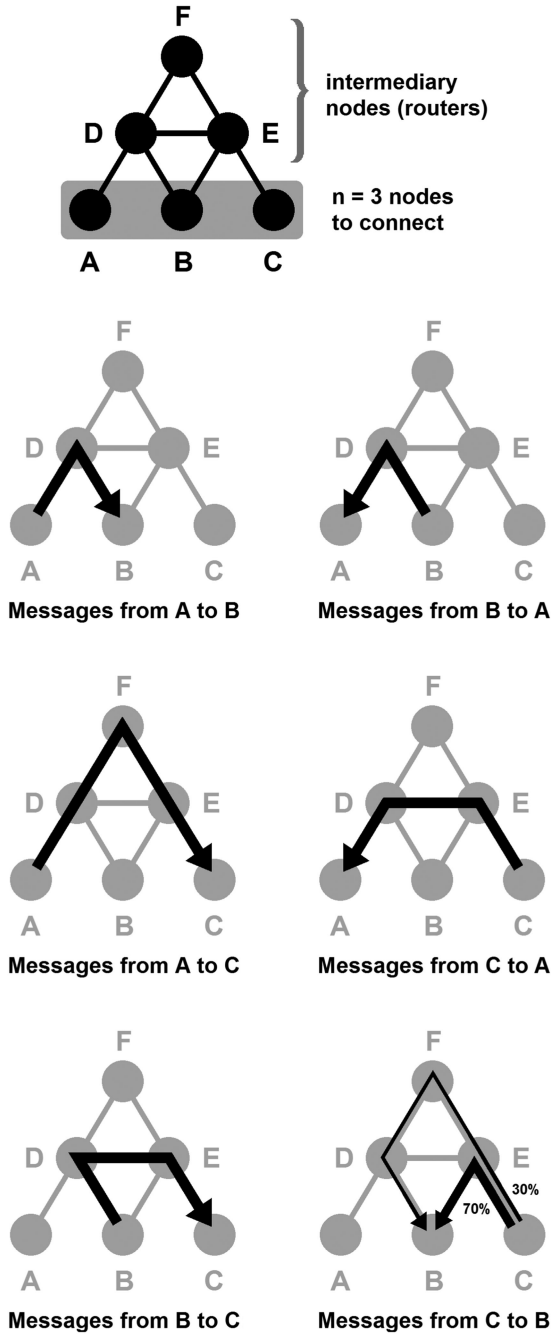

Fig. 1. In this graph (toy example), $n = 3$ nodes A, B and C are connected by 3 intermediary nodes D, E and F ($S = \{A, B, C\}$ here). The pictures describe the (arbitrary) paths used by the flow of messages from any node to any other node. The paths are not necessarily symmetrical: for instance, the path from A to C and the path from C to A are different. Besides, the flow of messages can be split into several paths: for the messages from C to B, 70 percent of the flow goes through (C, E, B), and 30 percent of the flow goes through (C, E, F, D, B). If we add up the flows of the six pictures, the maximal flow of messages is reached for node D.

3. We consider a continuous flow of messages, to abstract away the granularity of messages. This continuous flow of messages does not represent the network at a given instant, but rather the quantity of messages exchanged in a given time period, which is assumed to be relatively stable.

(which takes two nodes p and q of S as input, and returns a set of weighted paths in output). For instance, in the toy example of Fig. 1, $R(C,B) = \{(P_1, 0.7), (P_2, 0.3)\}$, with $P_1 = (C, E, B)$ and $P_2 = (C, E, F, D, B)$.

We say that a path $(u_1, \ldots, u_m)$ *crosses* a node p if there exists $i \in \{1, \ldots, m\}$ such that $u_i = p$. Similarly, we say that this path *crosses* a channel $\{p, q\}$ if there exists $i \in \{1, \ldots, m-1\}$ such that $u_i = p$ and $u_{i+1} = q$. A weighted path (P, α) crosses a node or channel x if the path P crosses x. For a given node or channel x, we now define the *flow* of messages $f(x)$ crossing x. Let $\Omega = \bigcup_{\{p,q\} \subseteq S} R(p,q)$ be the set containing *all* weighted paths used by the nodes of S. Let $W = \{(Q_1, \beta_1), (Q_2, \beta_2), \ldots, (Q_k, \beta_k)\}$ be the set of weighted paths of Ω crossing x. Then, $f(x) = (\beta_1 + \beta_2 + \ldots + \beta_n)F/(n-1)$ (the sum of the flows of messages crossing x). The *maximal flow* of (G, S, R) is $f_{\max} = \max_{(x \in V) \vee (x \in E)} f(x)$ (the maximal flow crossing a node or channel of G).

Generalized Fluid Message Flow (GFMF). We generalize the previous model to take failures into account. Here, R_n now takes two additional parameters $\mathcal{V}$ and $\mathcal{E}$, where $\mathcal{V}$ (resp. $\mathcal{E}$) represents the set of faulty nodes (resp. channels)—that is, the routing map adapts to the failures of nodes and channels in order to find correct paths, when it is possible. Thus, a set of weighted paths $R_n(p,q)$ becomes $R_n^{\mathcal{V},\mathcal{E}}(p,q)$, and the routing map R_n becomes $R_n^{\mathcal{V},\mathcal{E}}$. If this set of paths does not contain any faulty node or channel, we say that p and q are *reliably connected*. We will first consider faults as crashes for simplicity of presentation and then, later, we will discuss Byzantine failures.

3 Problems

The parameters λ (failure rate) and μ (communication probability) defined in Section 2 are fixed constants of the following problems.

The WRBD (Weak Reliable Bounded Degree) problem consists in finding, for any $n \geq 2$, a graph G_n satisfying the three following properties:

1) *Reliability.* Assume each node and channel crashes with probability at most λ (the probabilities being independent). Then, there exists a set S_n of n nodes of G_n such that any two correct nodes of S_n are connected with probability at least μ.
2) *Bounded degree.* There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.
3) *Linear number of nodes.* There exists a constant C such that, $\forall n \geq 2$, the number of nodes of G_n is at most Cn.

The RBD (Reliable Bounded Degree) problem consists in finding, for any $n \geq 2$, a graph G_n containing *exactly* n nodes and satisfying the two following properties:

1) *Reliability.* Assume each node and channel crashes with probability at most λ (the probabilities being independent). Then, any two correct nodes of G_n are connected with probability at least μ.
2) *Bounded degree.* There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.

The BDF (Bounded Degree and Flow) problem considers the FMF model (of Section 2) and consists in finding, for any $n \geq 2$, a tuple (G_n, S_n, R_n)—where G_n is a graph, S_n is a set of n nodes of G_n, and R_n is a routing map of S_n—satisfying the two following properties:

1) *Bounded Degree.* There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.
2) *Bounded Flow.* There exists a constant f_0 such that, $\forall n \geq 2$, the maximal flow of (G_n, S_n, R_n) is at most f_0.[4]

The RBDF (Reliable Bounded Degree and Flow) problem considers the GFMF model (of Section 2) and consists in finding, for any $n \geq 2$, a tuple $(G_n, S_n, R_n^{\mathcal{V},\mathcal{E}})$—where $G_n = (V_n, E_n)$ is a graph, S_n is a set of n nodes of G_n, and $R_n^{\mathcal{V},\mathcal{E}}$ is a routing map of S_n – satisfying the three following properties:

1) *Bounded Degree.* There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.
2) *Bounded Flow.* There exists a constant f_0 such that, $\forall n \geq 2$, $\forall \mathcal{V} \subseteq V_n$ and $\forall \mathcal{E} \subseteq E_n$, the maximal flow of $(G_n, S_n, R_n^{\mathcal{V},\mathcal{E}})$ is at most f_0.
3) *Reliability.* Assume each node and channel crashes with probability at most λ (the probabilities being independent). Let $\mathcal{V}$ (resp. $\mathcal{E}$) be the set of crashed nodes (resp. channels). Then, any two correct nodes of S_n are reliably connected in $R_n^{\mathcal{V},\mathcal{E}}$ with probability at least μ.

4 Solving the WRBD Problem

In this section, we provide a solution to the WRBD problem (Section 4.1) and prove its correctness (Section 4.2).

4.1 Solution

We describe here a graph G_n that solves the WRBD problem. We first give an overview, then the complete definition.

Overview. We first define the notion of *layered graph*, namely a graph where nodes are separated into several layers, and where only nodes of two adjacent layers can be connected. Then, we describe two layered graphs: T_n, which contains a binary tree connecting at least n nodes, and F_n, which is a recursive graph defined by induction. The recursive definition of F_n enables to preserve a constant communication probability between the first and last layer (independently of n) when $\lambda < 0.01$ (Lemma 1).[5] We show how to overcome this "$\lambda < 0.01$" constraint below. Besides, F_n is defined so that the number of nodes doubles at most every 2 layers, which enables to preserve a linear number of nodes, as shown in Theorem 3. The number of layers of T_n is adjusted so that T_n and F_n have the same number of layers H_n.

We consider a graph X_n, which is a layer-by-layer product of T_n and F_n, and a graph Y_n, which puts two graphs X_n in parallel. Doing so ensures a constant communication probability between any two nodes of the first layer.

We then apply three transformations in order to reach any communication probability μ with any failure rate λ. First, we connect several graphs Y_n in parallel, in order to achieve any communication probability μ. Second, we replicate each node, in order to simulate a failure rate $\lambda < 0.01$ for each node. Third, we replicate each channel, in order to

4. The "bounded flow" constraint here represents the capacity limitations of the network.

5. Note that this bound "$\lambda < 0.01$" is not supposed to be tight, and is simply small enough to have the desired property.

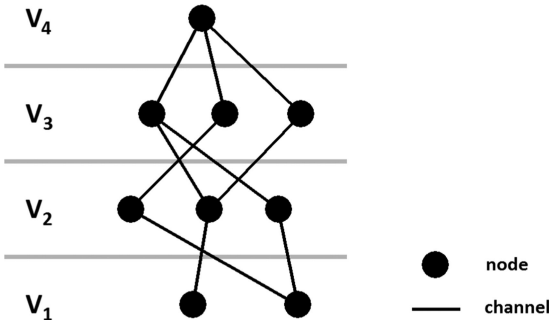

Fig. 2. A layered graph of height $H = 4$.

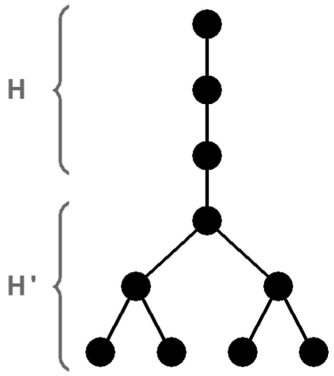

Fig. 3. Structure of graph T_m.

simulate a failure rate $\lambda < 0.01$ for each channel. The graph thus obtained is G_n.

Definitions.[6] We introduce a few variables below, then give a preliminary intuition of their use in the construction of the graph.

For any $n \geq 2$, let h_n be the smallest integer such that $2^{h_n-1} \geq n$. Let K_n be the smallest integer such that $2 + 4K_n \geq h_n$, and let $H_n = 2 + 4K_n$. Let α be the smallest integer such that $\alpha \geq 1$ and $0.5^\alpha \leq 1 - \mu$. Let β be the smallest integer such that $\beta \geq 1$ and $\lambda^\beta \leq 0.01$.

h_n is the height of a binary tree connecting at least n leaves. However, as our construction relies on an inductive process (see Fig. 11), The minimal height we can have is actually H_n. α corresponds to the number of replications of the graph, as shown in Fig. 6. β corresponds to the number of replications of each node and channel, as shown in.

A *layered graph* of height H is a tuple $(V_1, \ldots, V_H, E)$ satisfying the following conditions:

1) (V, E) is a graph with $V = \bigcup_{i \in \{1,\ldots,H\}} V_i$.
2) The sets V_i (layers) are disjoint: $\forall \{i, j\} \subseteq \{1, \ldots, H\}$, $V_i \cap V_j = \emptyset$.
3) The channels only connect neighbor layers: $\forall \{p, q\} \in E$, if $p \in V_i$ and $q \in V_j$, then $|i - j| = 1$.

An example of a layered graph is given in Fig. 2. By convention, in the following figures, V_1 always corresponds to the lower layer on the figure. We call V_1 the *first* layer and V_H the *last* layer.

6. These definitions are specific to the current graph. The same goes for the definitions in the following sections.

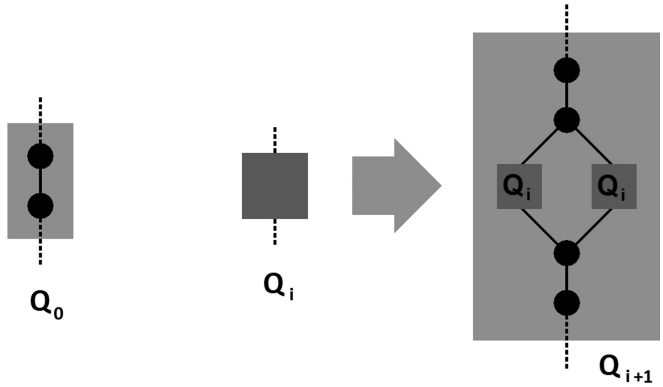

Fig. 4. Construction (by induction) of graph Q_i. The graph is defined so that the number of nodes doubles at most every 2 layers, which enables to preserve a linear number of nodes (see Theorem 3).

Graph T_n. We first define a tree-like layered graph of height H_n. Consider the layered graph represented in Fig. 3: this graph is composed of a line of height $H = 3$ and of a binary tree of height $H' = 3$. In other words, $\forall i \in \{1, \ldots, H'\}$, the layer i contains 2^{i-1} nodes, and the H remaining layers contain each 1 node. Then, $\forall n \geq 2$, we define T_n as a similar graph with $H = H_n - h_n$ and $H' = h_n$.

Graph F_n. $\forall k \geq 0$, we first define a layered graph Q_i by induction. Let Q_0 be a layered graph of height 2 containing 2 nodes and 1 channel, as described in Fig. 4. Then, $\forall i \geq 0$, Q_{i+1} is constructed with 2 instances of Q_i in parallel and 4 additional nodes, as described in Fig. 4 (Q_{i+1} has 4 more layers than Q_i). We now define F_n as follows: $\forall n \geq 2$, $F_n = Q_{K_n}$.

Graph X_n.[7] $\forall n \geq 2$, T_n is a layered graph of height H_n, and F_n is a layered graph of height $2 + 4K_n = H_n$. As T_n and F_n are layered graphs, let $T_n = (V_1, \ldots, V_{H_n}, E)$ and $F_n = (V'_1, \ldots, V'_{H_n}, E')$. Then, $\forall n \geq 2$, we define the layered graph $X_n = (V^*_1, \ldots, V^*_{H_n}, E^*)$ as follows:

- $\forall i \in \{1, \ldots, H_n\}$, to each pair of nodes (u, v) $nV_i \times V'_i$, we associate a unique node $p = f(u, v) \in V^*_i$ (thus, $|V^*_i| = |V_i||V'_i|$).
- Let $p = f(u, v)$ and $p' = f(u', v')$. Then, p and p' are neighbors in X_n if and only if u and u' (resp. v and v') are neighbors in T_n (resp. F_n).

Observe that, as the last layers of T_n and F_n contain 1 node, the last layer of X_n also contains 1 node.

Graph Y_n. $\forall n \geq 2$, we define graph Y_n as follows: we consider two instances of X_n (X^A_n and X^B_n), we merge the nodes of their first layers, and we merge the nodes of their last layers. This is illustrated in Fig. 5.

Graph G_n. $\forall n \geq 2$, graph G_n is finally obtained by applying three successive transformations to Y_n:

(1) *Transformation 1 (Network replication).* First, we connect α instances of Y_n by merging the nodes of their first layers, as illustrated in Fig. 6 for $\alpha = 3$.[8]

7. The definition of X_n looks like a Cartesian product, but it is not: we do not make the product of the whole graphs, but of each pair of layer separately.

8. More precisely, let $E_i = (u^i_1, \ldots, u^i_Q)$ be the first layer of the ith instance of Y_n (with $i \in \{1, \ldots, \alpha\}$). Then, $\forall j \in \{1, \ldots, Q\}$, we merge the α nodes $u^1_j, \ldots, u^\alpha_j$.

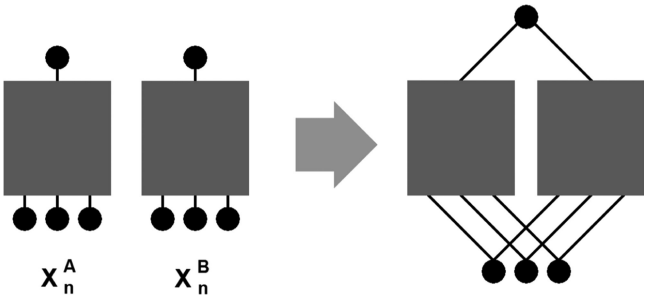

Fig. 5. Construction of graph Y_n.

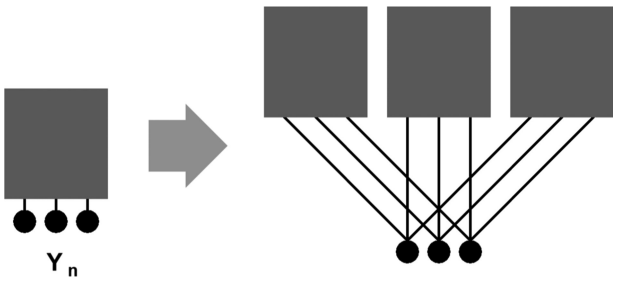

Fig. 6. Transformation 1 (Network replication) with $\alpha = 3$.

(2) *Transformation 2 (Node replication).* Second, we replace each node p by a set of β nodes $M(p)$. Then, for each channel $\{p, q\}$, we add a channel between each node of $M(p)$ and each node of $M(q)$ (see Fig. 7a).

(3) *Transformation 3 (Channel replication).* Third, we replace each channel by β channels in parallel (see Fig. 7b).

4.2 Correctness Proof

We prove that graph G_n described in Section 4.1 solves the WRBD problem. For this purpose, we prove the three properties of the WRBD problem: *Reliability*, *Bounded degree* and *Linear number of nodes*.

In Lemma 1, we show that, for a sufficiently small failure rate ($\lambda \leq 0.01$), the first layer and the last layer of F_n are connected with a constant probability (independently of n). To do so, we call P_i the probability that the first and last layer of Q_i are connected, then express P_{i+1} as a function of P_i (according to the inductive definition of Q_i). Then, we show that if $P_i \geq 0.8$, we also have $P_{i+1} \geq 0.8$. Thus, the first and last layer of Q_i (and thus, F_n) are connected with probability at least 0.8.

In Lemma 2, we show that the first layer of G_n contains at least n nodes. Then, we consider that S_n is a subset of the first layer of G_n to prove the following property.

In Theorem 1, we prove the *Reliability* property. We first consider the case $\lambda \leq 0.01$ and $\mu \leq 0.5$ (in this case, $Y_n = G_n$). According to the definition of X_n and Y_n, any two nodes of S_n are connected to the last layer of Y_n by two graphs F_n. Thus, the result, according to Lemma 2. We then consider that λ and μ can have any value, and show that the 3 final transformations of Section 4.1 enable to simulate the previous situation where $\lambda \leq 0.01$ and $\mu \leq 0.5$.

In Theorem 2, we prove the *Bounded degree* property. As G_n is intentionally defined as a combination of graphs with a bounded degree, the property follows.

In Theorem 3, we prove the *Linear number of nodes* property. We use the fact that the number of nodes of T_n is

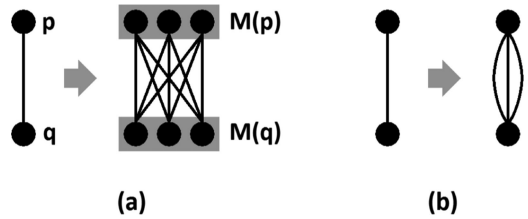

Fig. 7. Transformations 2 (Node replication) and 3 (Channel replication) with $\beta = 3$.

divided by 2 every layer (starting from the first layer), while the number of nodes of F_n at most doubles every 2 layers. Therefore, the number of nodes of X_n (which is the combination of T_n and F_n) is at least divided by 2 every 2 layers. Then, as $1 + 1/2 + 1/4 + 1/8 + \leq 2$, the number of nodes of X_n is linear in n, and so is the number of nodes of G_n.

Lemma 1. *Assume each node and channel crashes with probability at most λ (the probabilities being independent). If $\lambda \leq 0.01$, then $\forall n \geq 2$, the nodes of the first and last layer of F_n are both correct* and *connected with probability at least 0.8.*

Lemma 2. *$\forall n \geq 2$, the first layer of G_n contains at least n nodes.*

Theorem 1. *Assume each node and channel crashes with probability at most λ (the probabilities being independent). Then, there exists a set S_n of n nodes of G_n such that any two correct nodes of S_n are connected with probability at least μ.*

Theorem 2. *There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.*

Theorem 3. *There exists a constant C such that, $\forall n \geq 2$, the number of nodes of G_n is at most Cn.*

5 Solving the RBD Problem

In this section, we provide a solution to the RBD problem (Section 5.1) and prove its correctness (Section 5.2).

5.1 Solution

We describe here a graph G_n[9] that solves the RBD problem. We first give an overview, then the complete definition.

Overview. The idea is to combine several instances of a WRBD graph, each instance reliably connecting a smaller number of nodes, and to make their intermediary nodes disappear by merging them with other nodes.

Let W_m be any WRBD graph (for instance, the WRBD graph defined in Section 4). Then, $\forall n \geq 2$, we consider the largest integer m such that the number of nodes of W_m is at most n. If such a m does not exist, we define G_n as a complete graph with redundancy of channels. As it only happens for bounded values of n, it does not break the "Bounded degree" property.

Otherwise, we consider a set V of n nodes, and we split V into subsets of $\lfloor m/2 \rfloor$ nodes. Then, we connect each pair of subsets with an instance of W_m merged with the nodes of V. The resulting graph is G_n. Doing so ensures that any two nodes of V are reliably connected. Besides, according to the "Linear number of nodes" property of W_m, the number of instances of W_m is bounded, and so is the maximal degree of G_n.

9. The graph G_n of each section is different, each one solving one of the four problems.

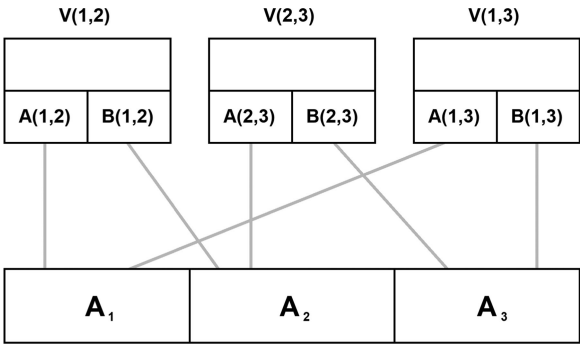

Fig. 8. Illustration of the construction of G_n for $M = 3$ (RBD problem). We represented $V(1,2)$, $V(2,3)$, $V(1,3)$ and V, as well as their subset $A(i,j)$, $B(i,j)$ and A_i. The grey lines show how the subsets $A(i,j)$ and $B(i,j)$ are merged with the subsets A_i.

Construction of G_n. Let $n \geq 2$, and let V be a set of n nodes.

Let W_m be any WRBD graph. Let N_m be the total number of nodes of W_m ($N_m \geq m$), and let S_m be the set of m nodes reliably connected by W_m.

If there exists no $m \geq 2$ such that $N_m \leq n$, then for any two nodes p and q of V, we add $\lceil \log(1-\mu)/\log(1-\lambda) \rceil$ channels between p and q (complete graph).

Otherwise, let $m \geq 2$ be the largest integer such that $N_m \leq n$. Let M be the smallest integer such that $M\lfloor m/2 \rfloor \geq n$. Let $\{A_1, \ldots, A_M\}$ be any set of M subsets of V such that $\bigcup_{i \in \{1,\ldots,M\}} A_i = V$ and $\forall i \in \{1, \ldots, M\}, |A_i| = \lfloor m/2 \rfloor$.

Then, $\forall \{i,j\} \subseteq \{1, \ldots, M\}$, we apply the following transformations. Let $W(i,j)$ be an instance of W_m, let $V(i,j)$ be the set of nodes of $W(i,j)$, and let $S(i,j)$ be the set of m nodes corresponding to S_m. Let $A(i,j)$ and $B(i,j)$ be two disjoint subsets of $S(i,j)$ such that $|A(i,j)| = |B(i,j)| = \lfloor m/2 \rfloor$. We merge the $\lfloor m/2 \rfloor$ nodes of $A(i,j)$ (resp. $B(i,j)$) with the $\lfloor m/2 \rfloor$ nodes of A_i (resp. A_j). Then, we merge the $N_m - 2\lfloor m/2 \rfloor$ nodes of $V(i,j) - A(i,j) - B(i,j)$ with any $N_m - 2\lfloor m/2 \rfloor$ nodes of $V - A_i - A_j$. The graph thus obtained is G_n. We illustrate this in Fig. 8.

5.2 Correctness Proof

We prove that graph G_n described in Section 5.1 solves the RBD problem. For this purpose, we prove the two properties of the WRBD problem: *Reliability* and *Bounded degree*.

In Theorem 4, we prove the *Reliability* property. Let p and q be two nodes of G_n. In the case where the graph is complete, the reliability property is ensured by the number of channels between p and q. Otherwise, it is ensured by the fact that p and q belong to the set S_m of at least one instance of W_m.

In Theorem 5, we prove the *Bounded degree* property. We first notice that the graph is complete only when $n \leq N_2$. Thus, in this case, the degree is bounded. Otherwise, we show that the number of subsets of $\lfloor m/2 \rfloor$ nodes is bounded (which is a consequence of the linearity property of the WRBD problem). Thus, the number of instances of W_m is bounded, and so is the degree of G_n.

Theorem 4. *Assume each node and channel crashes with probability at most λ (the probabilities being independent). Then, any two correct nodes of G_n are connected with probability at least μ.*

Theorem 5. *There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.*

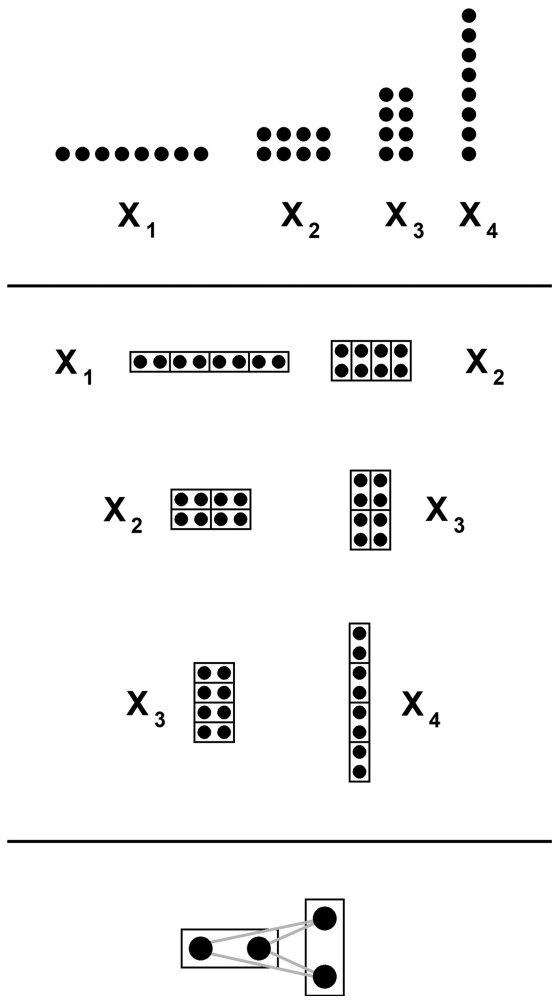

Fig. 9. Illustration of the construction of G_n for $H = 4$ (BDF problem). The first part represents the sets X_k for $k \in \{1,2,3,4\}$. The nodes $u_k(i,j)$ are ordered such that i (resp. j) corresponds to the horizontal (resp. vertical) axis. The second part shows how X_k and X_{k+1} are connected. X_k (resp. X_{k+1}) is partitioned into horizontal (resp. vertical) pairs of nodes. Each pair of nodes of X_k is connected to the pair of nodes of X_{k+1} with the corresponding position, as shown in the third part.

6 Solving the BDF Problem

In this section, we provide a solution to the BDF problem (6.1) and prove its correctness (6.2).

6.1 Solution

We describe here a tuple (G_n, S_n, R_n) that solves the BDF problem. We first give an overview, then the complete definition of G_n, S_n and R_n.

Overview. To construct G_n, the intuitive idea is the following. We define a sequence $(X_1, \ldots, X_H)$ of sets of $O(n)$ nodes. X_1, X_2, ..., X_H can be represented as tables of respectively $2^{H-1} \times 1$, $2^{H-2} \times 2$, ..., $1 \times 2^{H-1}$ nodes (each time, the width is divided by two and the height is multiplied by two). This is illustrated in Fig. 9. Then, each node of X_i is connected to two nodes of X_{i+1} with the same height modulo 2 and the same width modulo 2^{H-i}.[10] Finally, we

10. The demultiplexing properties of G_n are similar to those of a butterfly [20] network. However, G_n is defined differently. In a butterfly network, the nodes of each layer are described by an index i. Here, they are described by two indexes i and j ("$u_k(i,j)$").

merge X_1 and X_H so that the sets of nodes form a cycle. As we show further, this construction enables to mix the flows of messages in a perfectly balanced way. S_n is an arbitrary set of n nodes of the first layer of G_n.

We then define the routing map R_n as follows. The flows of messages between two nodes p and q of S_n take a unique path $r(p,q)$ (p is seen as a node of X_1 and q as a node of X_H). The path is determined by the binary decomposition of the position of q in X_H: at each new step, 0 means "go down" ($v_{k+1} = x(b_k)$) and 1 means "go up" ($v_{k+1} = y(b_k)$). This corresponds to the upper node and lower node in the third part of Fig. 9. We show that $r(p,q)$ actually reaches q in the correctness proof.

Graph G_n. Let H be the smallest integer such that $2^{H-1} \geq n$ (as $n \geq 2$, $H \geq 2$). We consider H sets of nodes $(X_1, \ldots, X_H)$, containing 2^{H-1} nodes each. $\forall k \in \{1, \ldots, H\}$, we denote each node of X_k by $u_k(i,j)$, with $i \in \{1, \ldots, 2^{H-k}\}$ and $j \in \{1, \ldots, 2^{k-1}\}$ (this is possible as $2^{H-k} \times 2^{k-1} = 2^{H-1}$). We connect these H sets of nodes with communication channels as follows. $\forall k \in \{1, \ldots, H-1\}$, $\forall i \in \{1, \ldots, 2^{H-k-1}\}$ and $\forall j \in \{1, \ldots, 2^{k-1}\}$, let $a = u_k(2i-1, j)$, $b = u_k(2i, j)$, $x = u_{k+1}(i, 2j-1)$ and $y = u_{k+1}(i, 2j)$. Then, we add the following communication channels: $\{a,x\}$, $\{a,y\}$, $\{b,x\}$ and $\{b,y\}$. Finally, $\forall i \in \{1, \ldots, 2^{H-1}\}$, we merge the node $u_1(i,1)$ with the node $u_H(1,i)$. The graph thus obtained is G_n.

Set of Nodes S_n. We define S_n as an arbitrary subset of the set X_1, containing exactly n nodes. This is possible as $2^{H-1} \geq n$.

Routing Map R_n. For a given node $v \in X_1 \cup \cup X_{H-1}$, let k, i and j be such that $v = u_k(i,j)$. Let i_0 be the smallest integer such that $2i_0 \geq i$. Let $x(v) = u_k(i_0, 2j-1)$ and $y(v) = u_k(i_0, 2j)$. Let $p \in X_1$ and $q \in X_H = X_1$. Let j be such that $q = u_H(1, j)$. Let $(b_1, \ldots, b_{H-1})$ be the binary sequence ($\forall k \in \{1, \ldots, H-1\}, b_k \in \{0,1\}$) such that $j - 1 = \Sigma_{k=1}^{k=H-1} b_k 2^{H-k-1}$ (that is, the binary decomposition of $i-1$).

Let $v_1 = p$. We define v_{k+1} by induction: if $b_k = 0$, $v_{k+1} = x(b_k)$, and if $b_k = 1$, $v_{k+1} = y(b_k)$. Let $r(p,q) = (v_1, \ldots, v_H)$. Then, we define the routing map R_n by $R_n(p,q) = \{(r(p,q), 1)\}$.

6.2 Correctness Proof

We prove that the tuple (G_n, S_n, R_n) described in Section 6.1 solves the BDF problem. For this purpose, we first prove that R_n is actually a routing map of S_n. Then, we prove the two properties of the BDF problem: *Bounded degree* and *Bounded flow*.

In Theorem 6, we show that R_n is a routing map of S_n. For this purpose, we show that the definition of $r(p,q)$ (with the binary decomposition of the position of q in X_H) is so that the path actually reaches q. To do so, we show by induction that the k first bits always reflect the position of the node crossed by $r(p,q)$ in X_k.

In Theorem 7, we prove the *Bounded degree* property: the degree of G_n is at most 4 by construction.

In Theorem 8, we prove the *Bounded flow* property: we show that according to the definition of the routing map, each node of X_k is crossed by $2^{k-1} \times 2^{H-k} = 2^{H-1}$ paths (which is a constant). Hence, the maximal flow is bounded.

Theorem 6. *R_n is a routing map of S_n.*

Theorem 7. *There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.*

Theorem 8. *There exists a constant f_0 such that, $\forall n \geq 2$, the maximal flow of (G_n, S_n, R_n) is at most f_0.*

7 Solving the RBDF Problem

In this section, we provide a solution to the RBDF problem (Section 7.1) and prove its correctness (Section 7.2).

7.1 Solution

We describe here a tuple $(G_n, S_n, R_n^{\mathcal{V},\mathcal{E}})$ that solves the RBDF problem. We first give an overview, then the complete definition of G_n, S_n and $R_n^{\mathcal{V},\mathcal{E}}$.

Overview. Let G_n^0 be the BDF graph defined in Section 6.1. After introducing preliminary definitions, we first define graph G_n. For this purpose, we define 4 intermediary graphs A_n, F_n, P_n and X_n. All these graphs are *layered graphs*, as introduced in Section 4.1, and have the same height H'_n. A_n is an variation of the previous graph G_n^0 with additional layers. F_n is a recursive graph designed to satisfy the reliability property. P_n is an adaptation of F_n to the reliability parameters λ and μ. Similarly to Section 4.1, X_n is a layer-by-layer product of A_n and P_n, in order to combine the properties of the previous graph G_n^0 with the reliability property of P_n. G_n is finally obtained by merging the first and the last layer of X_n, similarly to G_n^0. S_n is an arbitrary set of n nodes of the first layer of G_n.

To define routing map $R_n^{\mathcal{V},\mathcal{E}}$, the intuitive idea is the following. For any two nodes p and q of S_n, we first define a subgraph $W(p,q)$. Schematically, if p' and q' are the two corresponding nodes in G_n^0, and $r(p',q')$ is the path connecting them, then $W(p,q)$ is the instance of B_n corresponding to $r(p',q')$ in G_n. Then, the routing map connects p and q with a unique path avoiding the crashed nodes and channels in $W(p,q)$ (if it exists).

Definitions. Let $\epsilon > 0$ be any arbitrary positive constant. ϵ is the constant determining the cost of the graph (in terms of number of components). Therefore, it impacts many subsequent parameters.

Let K be the smallest integer such that $K \geq 2^{1/\epsilon}$. K is a parameter involved in the definition of graph F_n. $\forall n \geq 2$, let H_n be the smallest integer such that $2^{H_n - 1} \geq n$. We define the following sequence $(h_0, h_1, h_2, \ldots)$ by induction: $h_0 = 1$, and $\forall i \geq 0$, $h_{i+1} = 2 + Kh_i$. $\forall n \geq 2$, let M_n be the smallest integer such that $h_{M_n} \geq H_n$. Let $H'_n = h_{M_n}$. H'_n corresponds to the height of the layers graphs A_n, F_n, B_n, X_n and G_n.

Let $g(x) = 2x^K - x^{2K}$. Let z be the smallest integer such that $g(\gamma_z) \geq \gamma_z$, with $\gamma_z = 1 - (1/2^z)$ (we show that such an integer z always exists in Lemma 3 in Section 7.2), and let $\mu_0 = \gamma_z$. Let $\lambda_0 = \min(1 - \mu_0, 1 - (\mu_0 / g(\mu_0))^{1/(4+2K)})$. Let α be the smallest integer such that $\alpha \geq 1$ and $(1-\mu_0)^\alpha \leq 1 - \mu$. Let β be the smallest integer such that $\beta \geq 1$ and $\lambda^\beta \leq \lambda_0$. The parameters α and β impact the redundancy of nodes and channels in the definition of B_n.

Let (G_n^0, S_n^0, R_n^0) be the solution to the BDF problem described in Section 6.1.

Graph G_n. To define $G_n = (V_n, E_n)$, we first define 4 intermediary graphs A_n, F_n, B_n and X_n.

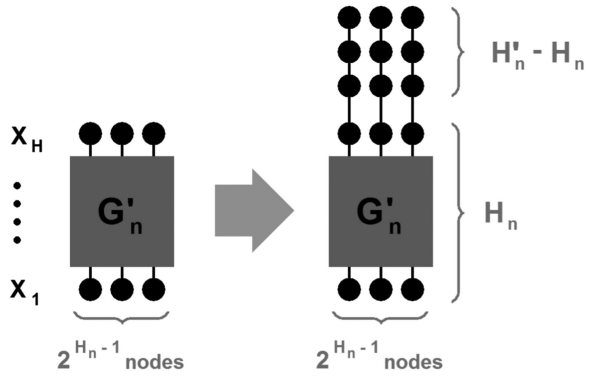

Fig. 10. Construction of graph A_n with graph G'_n and 2^{H_n-1} sequences of $h_{M_n} - H_n$ nodes.

$\forall n \geq 2$, we define layered graph A_n as follows. Consider graph G_n^0 and its definition in Section 6.1. The last step of the construction of G_n^0 consists in merging the nodes of X_1 and X_H. Let G'_n be graph G_n^0 just before this last step. Then, G'_n can be seen as a layered graph of height $H = H_n$, where the H layers are $(X_1, \ldots, X_H)$. We define graph A_n as a combination of G'_n and of 2^{H_n-1} sequences of $H'_n - H_n$ nodes, such as described in Fig. 10. Thus, A_n is a layered graph of height H'_n.

$\forall i \geq 0$, we first define a layered graph Q_i by induction. Let Q_0 be a layered graph of height 1 containing 1 node (see Fig. 11). Then, $\forall i \geq 0$, Q_{i+1} is constructed with $2K$ instances of Q_i and 2 additional nodes, as described in Fig. 11. We now define F_n as follows: $\forall n \geq 2$, $F_n = Q_{M_n}$.

$\forall n \geq 2$, graph B_n is obtained by applying three successive transformations to F_n. Transformation 1 consists in connecting α instances of F_n by merging the nodes of their first layers and then of their last layers. Transformations 2 and 3 are the same as for the WRBD graph.

$\forall n \geq 2$, A_n is a layered graph of height H'_n, and F_n is also a layered graph of height H'_n (by definition of H'_n). Thus, B_n is also a layered graph of height H'_n. As A_n and B_n are layered graphs, let $A_n = (V_1, \ldots, V_{H'_n}, E)$ and $B_n = (V'_1, \ldots, V'_{H'_n}, E')$. Then, $\forall n \geq 2$, we define the layered graph $X_n = (V''_1, \ldots, V''_{H'_n}, E'')$ by the same mechanism as for the WRBD graph.

The first layer V''_1 of X_n contains $m = 2^{H_n-1}$ nodes, and so does its last layer $V''_{H'_n}$. Let $V''_1 = \{u_1, \ldots, u_m\}$ and $V''_{H'_n} = \{v_1, \ldots, v_m\}$ (the order of numbering is unimportant here). We finally obtain graph G_n as follows: $\forall i \in \{1, \ldots, m\}$, we merge the nodes u_i and v_i.

Set of Nodes S_n. Let S'_n be any set of any n nodes of the first layer of X_n (such a set exists, as $|V''_1| \geq 2^{H_n-1} \geq n$). We define S_n as the corresponding set of nodes in G_n.

Routing Map $R_n^{\mathcal{V},\mathcal{E}}$. Let p and q be any two nodes of S_n. As G_n is obtained by merging the nodes of V''_1 and $V''_{H'_n}$ in X_n, let p'' (resp. q'') be the corresponding node if V''_1 (resp. $V''_{H'_n}$). According to the definition of X_n, let p_F (resp. q_F) be the node of A_n such that there exists a node v (resp v') such that $p'' = \pi(p_F, v)$ (resp. $q'' = \pi(q_F, v')$). According to the definition of A_n, let p_G be the node of G'_n corresponding to p_F, and let q_G be the node of the last layer of G'_n which is connected to q_F by a path of $H'_n - H_n$ nodes (according to Fig. 10). Finally, let p' (resp. q') be the node corresponding to p_G (resp. q_G) in G_n^0.

Let $r(p', q')$ be the path connecting p' and q' in G_n^0, such as defined in Section 6.1 (as shown in the proof of Theorem 6, $r(p', q')$ actually connects p' and q'). Let $r_G(p_G, q_G)$ be the corresponding path in G'_n. Let $r_F(p_F, q_F) = (u_1, \ldots, u_{H'_n})$ be an extension of $r_G(p_G, q_G)$ connecting p_F and q_F in A_n with $H'_n - H_n$ additional nodes (see Fig. 10). $\forall i \in \{1, \ldots, H'_n\}$, let W_i be the set of nodes w of X_n such that there exists a node v such that $w = \pi(u_i, v)$. Let $W = \bigcup_{i \in \{1,\ldots,H'_n\}} W_i$. Let W' be the corresponding set of nodes in G_n. We define $W(p, q)$ as the subgraph containing the nodes of W (and the channels connecting them) in G_n.

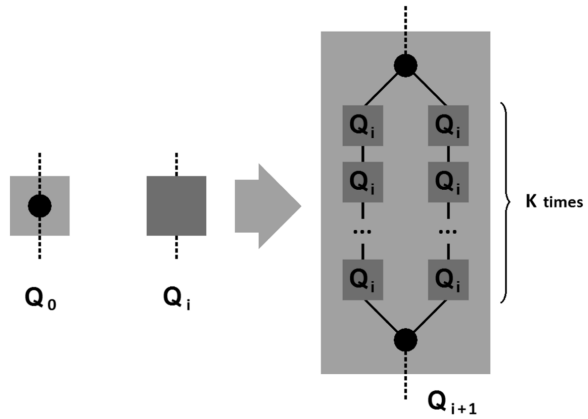

Fig. 11. Construction (by induction) of graph Q_i.

Now, let $\mathcal{V}$ (resp. $\mathcal{E}$) be any arbitrary set of crashed nodes (resp. edges) of G_n. If there exists a path of correct nodes and channels connecting p and q in $W(p, q)$, let $\psi(\mathcal{V}, \mathcal{E}, p, q)$ be this path. Otherwise, let $\psi(\mathcal{V}, \mathcal{E}, p, q)$ be any path connecting p and q in $W(p, q)$. We define the routing map R_n by $R_n^{\mathcal{V},\mathcal{E}}(p, q) = \{(\psi(\mathcal{V}, \mathcal{E}, p, q), 1)\}$ for any two nodes p and q of S_n.

7.2 Correctness Proof

We prove that the tuple $(G_n, S_n, R_n^{\mathcal{V},\mathcal{E}})$ described in Section 7.1 solves the RBDF problem. For this purpose, we prove the three properties of the RBDF problem: *Bounded degree*, *Bounded flow* and *Reliability*.

In Lemma 3, we prove a small property assumed in the description of the RBDF solution in Section 7.1.

In Theorem 9, we show the *Bounded degree* property, which follows from the construction of the graph.

In Theorem 10, we show the *Bounded flow* property: the worst case in terms of maximal flow (after merging several nodes) corresponds to our solution to the BDF problem.

In Lemma 4, we show that if the failure rate is at most λ_0, then the communication probability in Q_i (and thus, in F_n) is at least μ_0. This is due to the recursive definition of Q_i, which enables this property to propagate through each recursive step. In Lemma 5, we show that the three transformations between F_n and B_n adapt the result of Lemma 4 to any parameters λ and μ. Then, in Theorem 11, we show the *Reliability* property, which follows from the properties of B_n.

Lemma 3. *Let $\gamma_i = 1 - (1/2^i)$. There exists an integer $i \geq 1$ such that $g(\gamma_i) \geq \gamma_i$.*

Theorem 9. *There exists a constant Δ such that, $\forall n \geq 2$, the maximal degree of G_n is at most Δ.*

Theorem 10. *There exists a constant f_0 such that, $\forall n \geq 2$, $\forall \mathcal{V} \subseteq V_n$ and $\forall \mathcal{E} \subseteq E_n$, the maximal flow of $(G_n, S_n, R_n^{\mathcal{V},\mathcal{E}})$ is at most f_0.*

Lemma 4. *$\forall i \geq 0$, let p_i (resp. q_i) be the node of the first (resp. last) layer of Q_i. Suppose that $\lambda \leq \lambda_0$. If each node and channel crashes with probability at most λ, p_i and q_i are connected with probability at least μ_0.*

Lemma 5. *$\forall n \geq 2$, let p_n (resp. q_n) be any node of the first (resp. last) layer of B_n. If p_n and q_n are correct, and each other node and channel crashes with probability at most λ, then p_n and q_n are connected with probability at least μ.*

Theorem 11. *Assume each node and channel crashes with probability at most λ (the probabilities being independent). Let $\mathcal{V}$ (resp. $\mathcal{E}$) be the set of crashed nodes (resp. channels). Then, any two correct nodes of S_n are reliably connected in $R_n^{\mathcal{V},\mathcal{E}}$ with probability at least μ.*

8 Cost

In this section, we show that solving the BDF problem requires at least $\Omega(n \log n)$ nodes (8.1), then that our solution to the BDF and RBDF problems contain respectively $O(n \log n)$ and $O(n \log^{1+\epsilon} n)$ nodes (Sections 8.2 and 8.3).

8.1 Lower Bound on the BDF Problem

In Theorem 12, we show that solving the BDF problem requires at least $\Omega(n \log n)$ nodes.

In broad outline, we assume a solution (G_n, S_n, R_n) of the BDF problem. We first show that there are at least $\Omega(n^2)$ tuples of nodes (p, q) of S_n such that p and q are at distance at least $\Omega(\log n)$ from each other, due to the bounded degree. Therefore, as the flow of messages sent by each node of S_n is divided between the $n-1$ other nodes, the sum of the flows of all nodes is $\Omega(n \log n)$. Thus, for the maximal flow to be bounded, at least $\Omega(n \log n)$ nodes are required.

Theorem 12. *A graph solving the BDF problem, if it exists, contains at least $\Omega(n \log n)$ nodes.*

8.2 Cost of our BDF Solution

In Theorem 13, we show that graph G_n described in Section 6.1 contains $O(n \log n)$ nodes: G_n is composed of H sets $(X_1, \ldots, X_H)$ of $O(n)$ nodes each, with $H = O(\log n)$.

Theorem 13. *Graph G_n, described in Section 6.1, contains $O(n \log n)$ nodes.*

8.3 Cost of our RBDF Solution

We show that graph G_n described in Section 7.1 contains $O(n \log^{1+\epsilon} n)$ nodes. In Lemma 6, we show that the layers of F_n contain $O(\log^{\epsilon} n)$ nodes. In Lemma 7, we show that the height of G_n is $O(\log n)$. Then, as shown in Theorem 14, G_n contains $O(\log^{\epsilon} n) \times O(\log n) \times O(n) = O(n \log^{1+\epsilon} n)$ nodes.

Lemma 6. *There exists a constant C_1 such that the layers of graph F_n contain at most $C_1 \log^{\epsilon} n$ nodes each.*

Lemma 7. *There exists a constant C_2 such that $H'_n \leq C_2 \log n$.*

Theorem 14. *Graph G_n, described in 7.1, contains $O(n \log^{1+\epsilon} n)$ nodes.*

9 Related Works

The area of robust network design is a vast domain. We thus focus of papers where the general objective is to build a graph with good connectivity properties.

A lot of work in distributed computing has been devoted to tolerating a specific number of failures [9], [11], [28]. A constant failure rate raises different problems when the size of the network is unbounded, e.g., even a very small failure rate can entirely change asymptotic properties.

In [15], [25], [27], random failures are considered, but the reliability criteria is that the *whole* graph should remain connected. In other words, for a failure rate λ and a maximal degree Δ, the probability that a single node is disconnected is at least λ^{Δ}, and the probability that the graph remains fully connected is at most $1 - \lambda^{\Delta}$. Thus, it is impossible to have an arbitrarily high reliability μ (which is required in the very definition of our problems).

In [7], [12], [26], the focus was on constructing a graph satisfying certain topological properties. In [12] and [7] however, the node degree is not bounded. In [26], the degree is bounded, and the reliability criteria is the *connectivity* of the graph – i.e., the number k of disjoint paths between two given nodes. However, the length of these paths increases with the number of nodes. Therefore, when each node or channel has a given probability to fail, the probability that the k paths are cut approaches 1. Thus, no bound can be given on the communication probability when each node and channel has a given probability of failure.

A lot of network topologies that were proposed to reliably connect a large number of nodes with a reasonable degree [10], [13], [16], [17], [23], [24], [29], [31] were empirical and have only been experimented through simulations: their performances were evaluated only for a specific number of nodes. In [13], [23], [24], [31], traffic congestion slowly increases when the size of the network increases. In [10], [16], [17], [29], if we consider the asymptotic behavior of the proposed graphs (i.e., when the number of nodes grows), either the communication probability approaches zero, or the maximal degree approaches infinity.

For the RBD problem, our approach was to construct a specific graph (step by step) to match the desired properties. Intuitively, another idea could be to use expander graphs [14], [18], [21]. However, solving the RBD problem with expander graphs may be harder than it seems, if not impossible. We discuss this below.

We would define a graph as a (K, A) vertex expander if, for any set S of at most K nodes, the nodes of S are connected to at least $A|S|$ nodes [4]. By definition, K is at most n/A (where n is the number of nodes).

One could then have the following intuition of proof. Let G be the graph, and let G' be the graph after removing all crashed nodes and channels. Let u be a node of G', and let S_i be the ith neighborhood of u in G'. Then, with a constant probability, we can show that $|S_{i+1}| \geq 2|S_i|$ (assuming that A is large enough). Thus, by induction, one could deduct that u is connected to a majority of the nodes of G' with a constant probability.

There is, however, at least one flaw is this reasoning.

First, to have $|S_{i+1}| \geq 2|S_i|$, we must have at least $|S_i|$ *correct* nodes connected to S_i by one *correct* channel. These nodes represent an average fraction $(1-\lambda)^2$ of the neighbors of S_i in G. Thus, we must have $A \geq 1/(1-\lambda)^2$. As $K \leq n/A$, $K \leq n(1-\lambda)^2$.

Second, to show that $|S_{i+1}| \geq 2|S_i|$, we have to use the property according to which S_i is connected to at least $A|S_i|$

nodes. However, the property only applies for $|S_i| \leq K$. Thus, all the sets S_i combined contain at most $2K + K + K/2 + K/4 + \leq 4K \leq 4n(1-\lambda)^2$ nodes.

Therefore, for some values of λ (e.g., when $(1-\lambda)^2$ is much smaller than $(1-\lambda)/2$), the sets S_i do not cover a majority of correct nodes with a constant probability. Note that increasing A does not help here.

In a nutshell, an idea could be to build a subgraph (with some reliability properties) around the initial node and the final node, until the two subgraphs intersect. However, this requires to use a property which is not satisfied for some failure rates, once the subgraph reaches a certain size. Thus, it cannot be proved that the two subgraphs intersect.

Whether or not the RBD problem can be solved using expanders graphs (or another family of graphs) remains an open problem. Such a claim would require a fully consistent proof, which does not exist to our knowledge. The contribution of this paper is to show that the RBD problem can be solved, the very nature of solution itself being secondary.

10 Concluding Remarks

The asymptotic behavior of a distributed system has been studied in the literature of distributed computing so far as a function of its number of nodes n. The parameters studied have typically been the message and memory complexities. Here, we consider, for the first time, the asymptotic *reliability* of the distributed system (i.e., the probability that any two nodes remain connected) and consider as a parameter the number of physical components needed to build the system. We show that it is possible to connect an arbitrarily large number of nodes with any desired level of reliability while preserving a bounded degree and a bounded throughput.

Our approach suggests several research directions. For instance, instead of considering a continuous flow of messages, we could model more accurately the granularity of messages with a probabilistic model. One could also consider the cost of physically wiring the network, and try to bound it.

Acknowledgments

This work has been supported in part by the European ERC (Grant 339539 - AOC) and by the Swiss National Science Foundation (Grant 200021_169588 TARBDA).

References

[1] CERN Computing. [Online]. Available: http://home.cern/about/computing
[2] Microsoft: Datacenter Growth Defies Moore's Law. [Online]. Available: http://www.pcworld.com/article/130921/article.html
[3] Microsoft Now has One Million Servers. [Online]. Available: http://tinyurl.com/millionservers
[4] Salil Vadhan. Expander Graphs. [Online]. Available: https://people.seas.harvard.edu/ salil/pseudorandomness/expanders.pdf
[5] The Blue Brain Project. [Online]. Available: http://bluebrain.epfl.ch/page-58110-en.html
[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, pp. 50–58, 2010.
[7] R. Baldoni, S. Bonomi, L. Querzoni, and S. T. Piergiovanni, "Investigating the existence and the regularity of logarithmic harary graphs," *Theoretical Comput. Sci.*, vol. 410, no. 21–23, pp. 2110–2121, 2009. doi:10.1016/j.tcs.2009.01.041.
[8] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," in *Synthesis Lectures on Computer Architecture*, 2nd ed. San Rafael, CA, USA: Morgan & Claypool, 2013.
[9] C. J. Colbourn, *The Combinatorics of Network Reliability*, vol. 200. New York, NY, USA: Oxford Univ. Press, 1987.
[10] P. Costa, A. Donnelly, G. O'Shea, and A. Rowstron, "CamCubeOS: A key-based network stack for 3D torus cluster topologies," in *Proc. 22nd Int. Symp. High-Perform. Parallel Distrib. Comput.*, 2013, pp. 73–84.
[11] D. Dolev, "The Byzantine generals strike again," *J. Algorithms*, vol. 3, no. 1, pp. 14–30, 1982.
[12] R. Friedman, S. Manor, and K. Guo, "Scalable stability detection using logical hypercube," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 9, pp. 972–984, Sep. 2002. doi:10.1109/TPDS.2002.1036070.
[13] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, "Scalable hybrid wireless network-on-chip architectures for multicore systems," *IEEE Trans. Comput.*, vol. 60, no. 10, pp. 1485–1502, Oct. 2011. doi:10.1109/TC.2010.176.
[14] D. Gillman, "A chernoff bound for random walks on expander graphs," *SIAM J. Comput.*, vol. 27, no. 4, pp. 1203–1220, 1998.
[15] O. Goldschmidt, P. Jaillet, and R. Lasota, "On reliability of graphs with node failures," *Netw.*, vol. 24, no. 4, pp. 251–259, 1994.
[16] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 51–62.
[17] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: A scalable and fault-tolerant network structure for data centers," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2008, pp. 75–86.
[18] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin Amer. Math. Soc.*, vol. 43, no. 4, pp. 439–561, 2006.
[19] P. Jalote, *Fault Tolerance in Distributed Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1994.
[20] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *Proc. 40th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2007, pp. 172–182. doi:10.1109/MICRO.2007.29.
[21] J. Kleinberg and R. Rubinfeld, "Short paths in expander graphs," in *Proc. 37th Annu. Symp. Foundations Comput. Sci.*, 1996, pp. 86–95.
[22] L. Lamport, R. E. Shostak, and M. C. Pease, "The Byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, 1982.
[23] J. Li, C. Blake, D. S.J. De Couto, H. Imm Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw.*, 2001, pp. 61–69. doi:10.1145/381677.381684.
[24] Y. C. Liang, Y. Zeng, E. C. Y. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 4, pp. 1326–1337, Apr. 2008. doi:10.1109/TWC.2008.060869.
[25] S. Liu, K.-H. Cheng, and X. Liu, "Network reliability with node failures," *Netw.*, vol. 35, no. 2, pp. 109–117, 2000.
[26] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience," in *Proc. ACM SIGCOMM Conf. Appl. Technol., Architectures Protocols Comput. Commun.*, 2003, pp. 395–406. doi:10.1145/863955.863999.
[27] L. Mol, "On connectedness and graph polynomials," PhD thesis, Dalhousie Univ. Halifax, Halifax, Nova Scotia, 2016, https://dalspace.library.dal.ca/bitstream/handle/10222/71408/Mol-Lucas-PhD-Math-June-2016.pdf
[28] M. Nesterenko and S. Tixeuil, "Discovering network topology in the presence of Byzantine faults," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 12, pp. 1777–1789, Dec. 2009.
[29] R. Niranjan, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: A scalable fault-tolerant layer 2 data center network fabric," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 39–50.
[30] R. D. Schlichting and F. B. Schneider, "Fail-stop processors: An approach to designing fault-tolerant computing systems," *ACM Trans. Comput. Syst.*, vol. 1, no. 3, pp. 222–238, 1983.
[31] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992. doi:10.1109/9.182479.

Rachid Guerraoui is a professor with the École Polytechnique Fédérale de Lausanne. He has been affiliated with the Research Center of Ecole des Mines de Paris, the Commissariat a l'Energie Atomique in Saclay, Hewlett-Packard Laboratories in Palo Alto and the Massachusetts Institute of Technology. His research is devoted to concurrent and distributed computing, from multiprocessors to wide-area networks.

Alexandre Maurer received the graduated degree from Ecole Normale Superieure de Cachan, and the PhD degree in computer science from Paris 6 Sorbonne University. He is a postdoctoral researcher with the École Polytechnique Fédérale de Lausanne. His research interests include fault tolerance, safe AI, and biological algorithms.