Brief Announcement: The Fault-Tolerant Cluster-sending Problem

Jelle Hellings
Exploratory Systems Lab, Department of Computer Science, University of California, Davis, USA
jhellings@ucdavis.edu

Mohammad Sadoghi
Exploratory Systems Lab, Department of Computer Science, University of California, Davis, USA
msadoghi@ucdavis.edu

Abstract

The development of fault-tolerant distributed systems that can tolerate Byzantine behavior has traditionally been focused on consensus protocols, which support fully-replicated designs. For the development of more sophisticated high-performance Byzantine distributed systems, more specialized fault-tolerant communication primitives are necessary, however.

In this brief announcement, we identify the *cluster-sending problem*—the problem of sending a message from one Byzantine cluster to another Byzantine cluster in a reliable manner—as such an essential communication primitive. We not only formalize this fundamental problem, but also establish lower bounds on the complexity of this problem under crash failures and Byzantine failures. Furthermore, we develop practical cluster-sending protocols that meet these lower bounds and, hence, have optimal complexity. As such, our work provides a strong foundation for the further exploration of novel designs that address challenges encountered in fault-tolerant distributed systems.

2012 ACM Subject Classification Theory of computation → Communication complexity; Theory of computation → Distributed algorithms

Keywords and phrases Byzantine clusters, message sending, lower bound, optimal protocol

Digital Object Identifier 10.4230/LIPIcs...

Related Version The technical report on which this brief announcement is based is available at `https://arxiv.org/abs/1908.01455`.

1 Introduction

Recently, the emergence of *blockchain technology* has fueled a renewed interest in the development of fault-tolerant distributed systems. The main focus of current developments is mostly limited to fully-replicated systems in which each participating replica has the same role. We envision the design and development of more sophisticated high-performance Byzantine systems in which replicas have specialized roles. An example of such a system would be a *sharded geo-scale design* in which data is kept in *local Byzantine clusters*. In such a sharded geo-scale design, many queries can efficiently be answered by involving only a single cluster [1, 2]. In this way, a sharded design will often improve scalability when dealing with massive large-scale databases. For answering more complex queries, we need cooperation between different clusters, however. Hence, to enable the design and development of such systems, we need reliable ways for Byzantine clusters to communicate and cooperate. We believe that the existing consensus protocols are insufficient to fulfill this aim: we can run a single global consensus protocol among all replicas in all clusters to enable sharing of data and queries, but this would be at high—*quadratic*—communication costs for all replicas involved and would eliminate any possible scaling benefits of a clustered design. Indeed, we believe that there is a pressing need for more specialized Byzantine communication primitives. In this announcement we formalize and study one such primitive, the *cluster-sending problem*.

2 On the Fault-Tolerant Cluster-sending Problem

A *cluster* $\mathcal{C}$ is a set of replicas. We write $\mathsf{f}(\mathcal{C}) \subseteq \mathcal{C}$ to denote the set of *faulty replicas* in $\mathcal{C}$ and $\mathsf{nf}(\mathcal{C}) = \mathcal{C} \setminus \mathsf{f}(\mathcal{C})$ to denote the set of *non-faulty replicas* in $\mathcal{C}$. We write $\mathbf{n}_\mathcal{C} = |\mathcal{C}|$, $\mathbf{f}_\mathcal{C} = |\mathsf{f}(\mathcal{C})|$, and $\mathbf{nf}_\mathcal{C} = |\mathsf{nf}(\mathcal{C})|$ to denote the number of replicas, faulty replicas, and non-faulty replicas in the cluster, respectively. We extend the notations $\mathsf{f}(\cdot)$, $\mathsf{nf}(\cdot)$, $\mathbf{n}_{(\cdot)}$, $\mathbf{f}_{(\cdot)}$, and $\mathbf{nf}_{(\cdot)}$ to arbitrary sets of replicas. A *cluster system* $\mathfrak{S}$ is a finite set of clusters such that communication between replicas in a cluster is *local* and communication between clusters is *non-local*. We assume that there is no practical bound on local communication, while global communication is limited, costly, and to be avoided. If $\mathcal{C}_1, \mathcal{C}_2 \in \mathfrak{S}$ are distinct clusters, then we assume that $\mathcal{C}_1 \cap \mathcal{C}_2 = \emptyset$: no replica is part of two distinct clusters.

▶ **Definition 1.** *Let $\mathfrak{S}$ be a system and $\mathcal{C}_1, \mathcal{C}_2 \in \mathfrak{S}$ be two clusters with non-faulty replicas ($\mathsf{nf}(\mathcal{C}_1) \neq \emptyset$ and $\mathsf{nf}(\mathcal{C}_2) \neq \emptyset$). The* cluster-sending problem *is the problem of sending a value v from $\mathcal{C}_1$ to $\mathcal{C}_2$ such that:* **(1)** *all non-faulty replicas in $\mathcal{C}_2$* receive *the value v;* **(2)** *only if all non-faulty replicas in $\mathcal{C}_1$* agree *upon sending the value v to $\mathcal{C}_2$ will non-faulty replicas in $\mathcal{C}_2$ receive v; and* **(3)** *all non-faulty replicas in $\mathcal{C}_1$ can* confirm *that the value v was received.*

In this announcement, we use the notation $i \operatorname{sgn} j$, with $i, j \geq 0$ and sgn the sign function, to denote i if $j > 0$ and 0 otherwise.

Lower bounds for the cluster-sending problem First, we consider systems with only *crash failures*, in which case we can lower bound the number of messages exchanged. This lower bound is entirely determined by the maximum number of messages that can get *lost* due to faulty replicas not sending messages or ignoring received messages. In situations in which some replicas need to send or receive *multiple* messages, the capabilities of faulty replicas to ignore messages is likewise multiplied. E.g., when the number of senders outnumbers the receivers, then some receivers must receive multiple messages. As these receivers could be faulty, this means they could cause loss of multiple messages. By a thorough analysis, we end up with the following lower bounds:

▶ **Theorem 2.** *Let $\mathfrak{S}$ be a system with crash failures, let $\mathcal{C}_1, \mathcal{C}_2 \in \mathfrak{S}$, and let $\{i, j\} = \{1, 2\}$ such that $\mathbf{n}_{\mathcal{C}_i} \geq \mathbf{n}_{\mathcal{C}_j}$. Let $q_i = (\mathbf{f}_{\mathcal{C}_i} + 1) \operatorname{div} \mathbf{nf}_{\mathcal{C}_j}$, let $r_i = (\mathbf{f}_{\mathcal{C}_i} + 1) \bmod \mathbf{nf}_{\mathcal{C}_j}$, and let $\sigma_i = q_i \mathbf{n}_{\mathcal{C}_j} + r_i + \mathbf{f}_{\mathcal{C}_j} \operatorname{sgn} r_i$. Any protocol that solves the cluster-sending problem in which $\mathcal{C}_1$ sends a value v to $\mathcal{C}_2$ needs to exchange at least σ_i messages.*

Next, we look at systems with *Byzantine failures and replica signing* (e.g., using digital signatures and a public-key cryptography infrastructure). In this environment, we prove a lower bound on the number of replica signatures (certificates) exchanged. In this case, the receiving cluster $\mathcal{C}_2$ must eventually receive $\mathbf{f}_{\mathcal{C}_1} + 1$ distinct certificates signed by distinct replicas in $\mathcal{C}_1$. Only after receipt of these $\mathbf{f}_{\mathcal{C}_1} + 1$ certificates can the replicas in $\mathcal{C}_2$ conclude that at least one of these certificates was sent by a non-faulty replica in $\mathcal{C}_1$. A thorough analysis reveals the following lower bounds:

▶ **Theorem 3.** *Let $\mathfrak{S}$ be a system with Byzantine failures and replica signing and let $\mathcal{C}_1, \mathcal{C}_2 \in \mathfrak{S}$. Consider the cluster-sending problem in which $\mathcal{C}_1$ sends a value v to $\mathcal{C}_2$.*

1. *Let $q_1 = (2\mathbf{f}_{\mathcal{C}_1} + 1) \operatorname{div} \mathbf{nf}_{\mathcal{C}_2}$, $r_1 = (2\mathbf{f}_{\mathcal{C}_1} + 1) \bmod \mathbf{nf}_{\mathcal{C}_2}$, and $\tau_1 = q_1 \mathbf{n}_{\mathcal{C}_2} + r_1 + \mathbf{f}_{\mathcal{C}_2} \operatorname{sgn} r_1$. If $\mathbf{n}_{\mathcal{C}_1} \geq \mathbf{n}_{\mathcal{C}_2}$, then any protocol that solves the cluster-sending problem needs to exchange at least τ_1 certificates.*
2. *Let $q_2 = (\mathbf{f}_{\mathcal{C}_2} + 1) \operatorname{div} (\mathbf{nf}_{\mathcal{C}_1} - \mathbf{f}_{\mathcal{C}_1})$, $r_2 = (\mathbf{f}_{\mathcal{C}_2} + 1) \bmod (\mathbf{nf}_{\mathcal{C}_1} - \mathbf{f}_{\mathcal{C}_1})$, and $\tau_2 = q_2 \mathbf{n}_{\mathcal{C}_1} + r_2 + 2\mathbf{f}_{\mathcal{C}_1} \operatorname{sgn} r_2$. If $\mathbf{n}_{\mathcal{C}_2} \geq \mathbf{n}_{\mathcal{C}_1}$, then any protocol that solves the cluster-sending problem needs to exchange at least τ_2 certificates.*

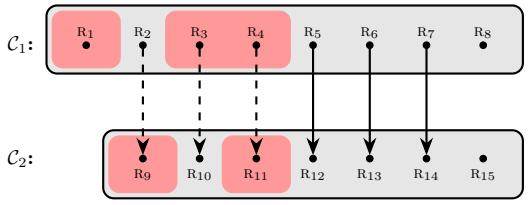

Figure 1 Bijection sending from $\mathcal{C}_1$ to $\mathcal{C}_2$. The faulty replicas are highlighted using a red background. The edges connect replicas $\mathrm{R} \in \mathcal{C}_1$ with $b(\mathrm{R}) \in \mathcal{C}_2$. Each solid edge indicates a message sent and received by non-faulty replicas. Each dashed edge indicates a message sent or received by a faulty replica.

Optimal cluster-sending via partitioned bijective sending We propose *bijective sending*, a powerful technique that allows the design of highly efficient cluster-sending protocols. In bijective sending, cluster $\mathcal{C}_1$ agrees on a value v. Then, the protocol chooses sets $S_1 \subseteq \mathcal{C}_1$ and $S_2 \subseteq \mathcal{C}_2$ of equal size and instruct each replica in $S_1 \subseteq \mathcal{C}_1$ to send v to a distinct replica in $\mathcal{C}_2$. By choosing S_1 and S_2 sufficiently large, we can guarantee successful cluster-sending. More specific, in the case of a system with *crash failures*, we need to choose $|S_1| = |S_2| = \mathbf{f}_{\mathcal{C}_1} + \mathbf{f}_{\mathcal{C}_2} + 1$. In the case of a system with *Byzantine failures* and replica signing, we need to choose $|S_1| = |S_2| = 2\mathbf{f}_{\mathcal{C}_1} + \mathbf{f}_{\mathcal{C}_2} + 1$. Next, we illustrate bijective sending

▶ **Example 4.** Let $\mathfrak{S}$ be a system with crash failures, let $\mathcal{C}_1 = \{\mathrm{R}_1, \ldots, \mathrm{R}_8\} \in \mathfrak{S}$ with $\mathsf{f}(\mathcal{C}_1) = \{\mathrm{R}_1, \mathrm{R}_3, \mathrm{R}_4\}$, and let $\mathcal{C}_2 = \{\mathrm{R}_9, \ldots, \mathrm{R}_{15}\} \in \mathfrak{S}$ with $\mathsf{f}(\mathcal{C}_2) = \{\mathrm{R}_9, \mathrm{R}_{11}\}$. We have $\mathbf{f}_{\mathcal{C}_1} + \mathbf{f}_{\mathcal{C}_2} + 1 = 6$. We choose $S_1 = \{\mathrm{R}_2, \ldots, \mathrm{R}_7\}$, $S_2 = \{\mathrm{R}_9, \ldots, \mathrm{R}_{15}\}$, and $b = \{\mathrm{R}_i \to \mathrm{R}_{i+7} \mid 2 \leq i \leq 7\}$.

In Figure 1, we sketched this situation. Replica R_2 sends a valid message to R_9. As R_9 is faulty, it might ignore this message. Replicas R_3 and R_4 are faulty and might not send a valid message. Additionally, R_{11} is faulty and might ignore any message it receives. The messages sent from R_5 to R_{12}, from R_6 to R_{13}, and from R_7 to R_{14} are all sent by non-faulty replicas to non-faulty replicas. Hence, these messages all arrive correctly.

The bijective sending techniques have optimal communication complexity. Unfortunately, bijective sending places unrealistic requirements on clusters that vastly differ in size. We can address this by *partitioning* the larger-sized cluster into a set of smaller clusters, and then letting sufficient of these smaller clusters participate independent in bijective sending. This approach results in the following:

▶ **Theorem 5.** *Let $\mathfrak{S}$ be a system and let $\mathcal{C}_1, \mathcal{C}_2 \in \mathfrak{S}$. Consider the cluster-sending problem in which $\mathcal{C}_1$ sends a value v to $\mathcal{C}_2$.*

1. *If $\mathbf{n}_{\mathcal{C}} > 3\mathbf{f}_{\mathcal{C}}$, $\mathcal{C} \in \mathfrak{S}$, and $\mathfrak{S}$ has crash failures, then we can use (partitioned) bijective sending as a solution to the cluster-sending problem with optimal message complexity. These protocols solve the cluster-sending problem using $\mathcal{O}(\max(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}))$ messages, of size $\mathcal{O}(\|v\|)$ each.*
2. *If $\mathbf{n}_{\mathcal{C}} > 4\mathbf{f}_{\mathcal{C}}$, $\mathcal{C} \in \mathfrak{S}$, and $\mathfrak{S}$ has Byzantine failures and replica sending, then we can use (partitioned) bijective sending as a solution to the cluster-sending problem with optimal replica certificate usage. These protocols solve the cluster-sending problem using $\mathcal{O}(\max(\mathbf{n}_{\mathcal{C}_1}, \mathbf{n}_{\mathcal{C}_2}))$ messages, of size $\mathcal{O}(\|v\|)$ each.*

References

1 M. Tamer Özsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Springer New York, 3th edition, 2011.

2 Maarten van Steen and Andrew S. Tanenbaum. *Distributed Systems*. Maarten van Steen, 3th edition, 2017. URL: `https://www.distributed-systems.net/`.