Brief Announcement: Effects of Topology Knowledge and Relay Depth on Asynchronous Consensus

Dimitris Sakavalas
Boston College, USA
dimitris.sakavalas@bc.edu

Lewis Tseng
Boston College, USA
lewis.tseng@bc.edu

Nitin H. Vaidya[1]
Georgetown University, USA
nitin.vaidya@georgetown.edu

Abstract

Consider an asynchronous incomplete directed network. We study the feasibility and efficiency of *approximate crash-tolerant* consensus under different restrictions on topology knowledge and relay depth, i.e., the maximum number of hops any message can be relayed.

2012 ACM Subject Classification Computer systems organization → Fault-tolerant network topologies

Keywords and phrases Asynchrony, crash fault, consensus, topology knowledge, relay

Digital Object Identifier 10.4230/LIPIcs.DISC.2018.51

Related Version A full version is available at [5], `https://arxiv.org/abs/1803.04513`.

1 Introduction

The fault-tolerant consensus problem introduced by Lamport et al. [4] and its variations have been studied extensively. The need to overcome the FLP impossibility result for consensus in asynchronous systems has led to the study of the *approximate consensus* problem [3], where nodes are required to output roughly the same value. We consider a directed network of n nodes, wherein at most f nodes are subject to crash failure. We explore the feasibility and efficiency of achieving approximate consensus in *asynchronous incomplete* networks under different restrictions on *topology knowledge* and *relay depth* (defined as the maximum number of hops that information can be propagated). These constraints are useful in large-scale networks to avoid memory overload and network congestion.

Our prior work [7] showed that exact crash-tolerant consensus is solvable in *synchronous* networks with only one-hop knowledge and relay depth 1, i.e., each node only needs to know its immediate neighbors, and no message needs to be relayed. Such a local algorithm is of practical interest due to low deployment cost and message complexity in each round. In *asynchronous* undirected networks, there exists a simple flooding-based algorithm adapted from [2] that achieves approximate consensus with up to f crash faults if the network satisfies $(f + 1)$

[1] This research is supported in part by National Science Foundation awards 1421918. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies or the U.S. government.

node-connectivity and $n > 2f$, where n is the number of nodes. However, the sufficiency of the conditions is not guaranteed if we restrict topology knowledge and relay depth. Motivated by this observation, this work addresses the following question in *asynchronous* systems:

What is a tight condition on the underlying communication graph to achieve approximate consensus if each node has only a k-hop topology knowledge and relay depth k'?

To the best of our knowledge, two prior papers [1, 6] examined a similar problem – *synchronous* Byzantine consensus. In [6], Su and Vaidya identified the condition under different relay depths. Alchieri et al. [1] studied the problem under unknown participants. The technique developed for asynchronous consensus in this work is significantly different. Please refer to our technical report [5] for more discussion on other related work.

Model and Terminology. The point-to-point message-passing network is represented by *directed* graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of n nodes, and $\mathcal{E}$ is the set of directed edges. The communication links are assumed to be reliable. Node i can transmit messages to its outgoing neighbors and itself. Up to f nodes may suffer crash failures in an execution, in which case they stop taking steps. We consider *asynchronous* communication. i.e., a message may be delayed arbitrarily but will eventually be delivered. Let N_i^-, N_i^+ denote the sets of incoming neighbors and outgoing neighbors of node i respectively. Also, for a node i, its *k-hop incoming neighbors* $N_i^-(k)$, are defined as the nodes j which can reach i using a directed path in G that has $\leq k$ hops. The notion of k-hop outgoing neighbors $N_i^+(k)$, is defined similarly. For set $B \subseteq \mathcal{V}$, node i is said to be an incoming neighbor of set B if $i \notin B$, and there exists $j \in B$ such that $(i, j) \in \mathcal{E}$. With N_B^- we will denote the incoming neighbors of B.

Approximate Consensus. In the approximate consensus problem [3], each node i maintains a *state* v_i with $v_i[p]$ denoting the state of node i at the end of phase (or iteration) p. The initial state of node i, $v_i[0]$, is equal to the initial input provided to node i. At the start of *asynchronous* phase p ($p > 0$), the state of node i is $v_i[p-1]$. Let $U[p]$ and $\mu[p]$ be the maximum and the minimum state at nodes that have not crashed by the end of phase p. Then, a *correct* approximate consensus algorithm needs to satisfy the following two conditions for any $\epsilon > 0$:

- *Validity*: $\forall p > 0, U[p] \leq U[0]$ and $\mu[p] \geq \mu[0]$; and
- *ϵ-Convergence*: $\exists p, \forall r \geq p, \quad U[r] - \mu[r] < \epsilon$.

2 Limited Topology Knowledge and Relay Depth

Prior works (e.g., [7]) assumed that each node has n-hop topology knowledge and relay depth n, which is not realistic in large-scale networks. Hence, we are interested in the family of algorithms (*iterative k-hop algorithms*) in which nodes only know their k-hop neighborhoods, and propagate state values to nodes that are at most k-hops away for $1 \leq k \leq n$. Note that no exchange of topology information takes place.

Iterative k-hop algorithms. Each node i performs the following three steps in phase p:

1. *Transmit*: Transmit messages of the form $(v_i[p-1], \cdot)$ to nodes that are reachable from node i via at most k hops away, through intermediate relays.
2. *Receive*: Receive messages from all k-hop incoming neighbors. Denote by $R_i[p]$ the set of messages that node i received at phase p.
3. *Update*: Update state using a transition function Z_i, where Z_i is a part of the specification of the algorithm, and takes as input the set $R_i[t]$. i.e., $v_i[t] := Z_i(R_i[t], v_i[t-1])$ at node i.

Main Results. Below, we present two definitions to facilitate the discussion.

▶ **Definition 1** ($A \rightarrow_k B$)**.** Given disjoint non-empty subsets of nodes A and B, we will say that $A \rightarrow_k B$ holds if there exists a node i in B for which there exist at least $f+1$ node-disjoint paths of length at most k from distinct nodes in A to i. More formally, if $\mathcal{P}_i^A(k)$ is the family of all sets of k-length node-disjoint paths (with i being their only common node) initiating in A and ending in node i, $A \rightarrow_k B$ means that $\exists i \in B, \max_{P \in \mathcal{P}_i^A(k)} |P| \geq f+1$.

▶ **Definition 2** (Condition k-CCA)**.** For any partition L, C, R of $\mathcal{V}$, where L and R are both non-empty, either $L \cup C \rightarrow_k R$ or $R \cup C \rightarrow_k L$.

▶ **Theorem 3.** *Approximate crash-tolerant consensus in an asynchronous system using iterative k-hop algorithms is feasible iff G satisfies Condition k-CCA.*

The complete proof is presented in [5]. We only sketch the proof here. The necessity is proved using an indistinguishable argument inspired by [3, 7]. For sufficiency, we present Algorithm k-LocWA. Our key contribution is to identify what are the set of messages that each node needs to receive before updating its state value in Step 3 of the iterative k-hop algorithms. Algorithm k-LocWA relies on *Condition k-WAIT*: For $F_i \subseteq N_i^-(k)$, we denote with $reach_i^k(F_i)$ the set of nodes that have paths of length $l \leq k$ to node i in G_{V-F_i}. That is, the set of k-hop incoming neighbors of i that remain connected with i even when all nodes in set F_i crash. The condition is satisfied at node i, in phase p if there exists $F_i \subseteq N_i^-(k)$ with $|F_i[p]| \leq f$ such that $reach_i^k(F_i[p]) \subseteq heard_i[p]$. Finally, we show that if G satisfies Condition k-CCA, then Algorithm k-LocWA correctly solves approximate consensus.

We derive an upper bound on the number of *asynchronous* phases needed for ϵ-convergence of Algorithm k-LocWA in [5]. This upper bound is naturally a function of values ϵ, k, f, n and $\delta = U[0] - \mu[0]$. As a function of k, the bound implies that for $k' \geq k$, Algorithm k'-LocWA ϵ-converges faster than k-LocWA. We also prove that for values $k, k' \in \mathbb{N}$ with $k \leq k'$, Condition k-CCA implies Condition k'-CCA and that n-CCA is equivalent to CCA from [7].

Topology Discovery and Unlimited Relay Depth. Even if the topology knowledge of the nodes is restricted to their 1-hop neighborhood, we show that allowing topology information exchange and relay depth n, one can achieve approximate consensus whenever condition CCA [7] holds. This can be achieved through Algorithm LWA, presented in the full version [5], which introduces a topology discovery mechanism to learn the crucial topology information that is necessary to achieve consensus. This result implies that knowledge of topology does not affect the feasibility of the problem if topology knowledge can be relayed.

References

1 Eduardo A. P. Alchieri, Alysson Neves Bessani, Joni da Silva Fraga, and Fabíola Greve. Byzantine consensus with unknown participants. In *OPODIS 2008*, volume 5401 of *LNCS*, pages 22–40. Springer, 2008. doi:10.1007/978-3-540-92221-6_4.

2 Danny Dolev. The Byzantine generals strike again. *Journal of Algorithms*, 3(1), 1982.

3 Danny Dolev, Nancy A. Lynch, Shlomit S. Pinter, Eugene W. Stark, and William E. Weihl. Reaching approximate agreement in the presence of faults. *J. ACM*, 33(3):499–516, 1986. doi:10.1145/5925.5931.

4 M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980. doi:10.1145/322186.322188.

5 Dimitris Sakavalas, Lewis Tseng, and Nitin H. Vaidya. Effects of topology knowledge and relay depth on asynchronous consensus. *CoRR*, abs/1803.04513, 2018. arXiv:1803.04513.

6 Lili Su and Nitin Vaidya. Reaching approximate Byzantine consensus with multi-hop communication. In *SSS 2015*, volume 9212 of *LNCS*, pages 21–35. Springer, 2015. `doi:10.1007/978-3-319-21741-3_2`.

7 Lewis Tseng and Nitin H. Vaidya. Fault-tolerant consensus in directed graphs. In *PODC '15*, pages 451–460. ACM, 2015. `doi:10.1145/2767386.2767399`.