S-3 Short Paper / Capacity of Byzantine Agreement: Summary of Recent Results*

Guanfeng Liang and Nitin Vaidya
Dept. of Electrical and Computer Engineering,
and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
{gliang2,nhv}@illinois.edu

ABSTRACT

In this paper, we consider the problem of maximizing the throughput of Byzantine agreement, for two cases: i. communication link's capacity is fixed; and ii. the sum capacity of all links in the system is fixed. Byzantine agreement is a classical problem in distributed computing, with initial solutions presented in the seminal work of Pease, Shostak and Lamport. The notion of throughput here is similar to that used in the networking/communications literature on unicast or multicast traffic. In case i, we characterize the maximum achievable agreement throughput in four-node networks. In case ii, we identify sufficient condition for achieving agreement throughput R.

Categories and Subject Descriptors

C.2.4 [**COMPUTER-COMMUNICATION NETWORKS**]: Distributed Systems

General Terms

Algorithms, Reliability, Security, Theory

Keywords

Byzantine, agreement, wireless

1. INTRODUCTION

We consider the problem of characterizing the capacity of Byzantine agreement, given the link capacity or sum capacity of the system is limited. Byzantine agreement is a classical problem in distributed computing, with initial solutions presented in the seminal work of Pease, Shostak and Lamport [1]. Many variations on the Byzantine *agreement* problem have been introduced in the past, with some of the variations also called *consensus*. We will use the following definition of Byzantine agreement: Consider a network with one node designated as the *sender* or *source* (S), and the other nodes designated as the *peers*. The goal of Byzantine agreement is for all the fault-free nodes to "agree on" the value being sent by the sender, despite the possibility that some of the nodes may be faulty. In particular, the following conditions must be satisfied:

- **Agreement:** All fault-free peers must agree on an identical value.
- **Validity:** If the sender is fault-free, then the agreed value must be identical to the sender's value.
- **Termination:** Agreement between fault-free peers is eventually achieved.

Our goal in this work is to design algorithms that can achieve maximum *throughput*, of agreement. When defining throughput, the "value" referred in the above definition of agreement is viewed as an infinite sequence of *information* bits. We assume that the information bits have already been compressed, such that for any subsequence of length $l > 0$, the 2^l possible sequences are sent by the sender with equal probability. Thus, no set of information bits sent by the sender contains useful information about other bits. This assumption comes from the observation about "typical sequences" in Shannon's work [3].

We also adopt the notion of *channel capacity* from the information theory literature [3]: tightest upper bound on the amount of *information* that can be reliably transmitted over a communications channel. Basically, for a link with capacity z bits/unit time, by definition of link *capacity*, at most z information bits can be "sent" per unit time - independent of how the bits are encoded (e.g. the bits could be encoded as a specific waveform, or as silenced interval). In the existing works on Byzantine agreement, the *capacity* of links between the nodes are assumed to be infinite implicitly. To the best of our knowledge, we are the first one to study the problem of Byzantine agreement when the links in the network have finite, and maybe different, capacity.

At each peer, we view the agreed information as being represented in an array of infinite length. Initially, none of the bits in this array at a peer have been agreed upon. As time progresses, the array is filled in with agreed bits. In principle, the array may not necessarily be filled sequentially. For instance, a peer may agree on bit number 3 before it is able to agree on bit number 2. Once a peer agrees on any bit, that agreed bit cannot be changed.

We assume that an agreement algorithm begins execution

*This research is supported in part by Army Research Office grant W-911-NF-0710287. 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.

S3'10, September 20, 2010, Chicago, Illinois, USA.

at time 0. The system is assumed to be synchronous. In a given execution of an agreement algorithm, suppose that by time t all the fault-free peers have agreed upon bits 0 through $b(t)-1$, and at least one fault-free peer has not yet agreed on bit number $b(t)$. Then, the agreement *throughput* is defined as $\lim_{t\to\infty} \frac{b(t)}{t}$.

Capacity of agreement in a given network, for a given sender and a given set of peers, is defined as the supremum of all achievable agreement throughputs.

2. FIXED LINK CAPACITY

We first consider the problem under the constraint that each point-to-point link in the system has a fixed finite capacity [2]. It is known that a network must contain at least 4 nodes for agreement to be achievable with a single Byzantine failure. In this work, we only consider the case of 4 nodes when at most 1 node may suffer Byzantine failure. The characterization of agreement capacity for the four-node network is non-trivial and cannot be generalized to larger networks directly. The design of capacity achieving algorithms in larger networks (possibly with multiple failures) is substantially more complex than the four-node case.

We consider a synchronous network of 4 nodes named S, A, B and C, with node S acting as the sender, and the others being the peers. At most one of these four nodes may be faulty. The network is viewed as a directed graph, formed by directed links between the nodes in the network, with the capacity of each link being finite. The capacity of some links may be 0, which implies that these links do not exist. Let us call the incoming links at S as the *uplinks* (links AS, BS and CS). We identify the following *necessary* conditions for achieving agreement throughput of R bits/unit time.

- **Necessary condition NC1:** If any one peer is removed from the network, the min-cut from the source S to each remaining peer must be at least R.
- **Necessary condition NC2:** The max-flow to each of the peers from the other peers, with the source removed from the network, must be at least R.
- **Necessary condition NC3:** All incoming links to the peers must exist (capacity>0).
- **Necessary condition NC4:** The capacity of every out-going link from S must be at least R, when there is no uplink.

Our main results are the tightness of these conditions:

- **With uplink(s):** Agreement capacity of a four-node network is the supremum over all throughputs R that satisfy necessary conditions NC1, NC2, and NC3.
- **With no uplink:** Agreement capacity of a four-node network is the supremum over all throughputs R that satisfy necessary conditions NC1, NC2, NC3 and NC4.

2.1 Sketch of Capacity Achieving Algorithms

We prove our results by providing agreement algorithms that can achieve throughput arbitrarily close to R, given the corresponding conditions are satisfied. The algorithm for the case of complete graphs is slightly different from the one for incomplete graphs, and is easier to describe. For brevity, we will only sketch the algorithm for the complete graph. Interested readers are referred to our technical report [2] for more details.

The proposed Byzantine agreement algorithm for the complete graph proceeds in rounds. The units for rate R and the various link capacities are assumed to be *bits/time unit*, for a convenient choice of the *time unit*. We assume that by a suitable choice of the *time unit*, the number R and the various link capacities can be turned into integers. The algorithm executes in multiple rounds, with the duration of each round being approximately c time units. Note that in c time units, a link with capacity z bits/time unit can carry z symbols (or packets) from Galois field GF(2^c). Computation is assumed to require 0 time.

In Round 1, the source S transmits as many coded packets as possible to the peers, such that each coded packet is a linear combination of R packets of data, and any subset of R coded packets constitutes **independent** linear combinations of the R data packets. As we know from the design of Reed-Solomon codes, if c is chosen large enough, this linear independence requirement can be satisfied. In round 2, each peer relays as many distinct packets it receives from S in round 1 to each of the other two peers. Then, each fault-free peer checks if any node has misbehaved by trying to find a unique solution for **each** subset of R packets from among the packets received from the other three nodes in rounds 1 and 2. We can show that if a faulty node misbehaves, it will be detected by at least one fault-free peer.

If a failure is detected, a broadcast phase is triggered, and every node including S **broadcasts** all packets it has sent and received during rounds 1 and 2 to the remaining 3 nodes, using the traditional Byzantine agreement algorithm, in particular the algorithm by Pease, Shostak and Lamport [1]. This is possible in the complete graphs. For incomplete graphs with fewer uplinks, this part is more complicated and is described in our technical report [2]. Based on the broadcast information, the fault-free nodes will be able to narrow down the location of the faulty node into a set containing at most 2 nodes. The operations after the first detection are similar to what has been described above, except that the schedule in round 2 may need to be modified depending on the narrowed down set of possibly faulty node. We show that the faulty node will be identified if it misbehaves for more than a finite number of times [2]. Once the faulty node is identified, each fault-free peer can recover the correct data from the packets from the other two fault-free nodes, or terminates the algorithm if S is faulty.

In achieving throughput R, it will be necessary to have multiple "generations" of packets in the network, with the algorithm operating in a pipelined manner (one round per pipeline stage). Agreement algorithm for one new generation of data of size Rc bits (or R symbols from GF(2^c)) starts per round. By using a suitably large c, the overhead for disseminating detection results and a finite number of broadcast phases diminishes to 0 as time goes to infinity. Hence, the throughput can be made arbitrarily close to R.

2.2 Discussion

While NC1 and NC2 can be easily generalized to larger networks with multiple failures, they are not sufficient for networks with more than 4 nodes. The following condition must also be satisfied for achieving throughput R:

- **Necessary condition NC5:** If any node is removed from the network, the sum capacity of all links in both direction on any cut must be at least R.

NC5 is implied by NC1 and NC2 in four-node networks, but not in larger networks.

3. FIXED SUM CAPACITY

The communication model we used in [2] assumes communications over different links are point-to-point, and will not interfere with each other, which is usually true in wired networks but not in wireless networks. The wireless medium has two main characteristics that differentiate it from the wire medium: **(1) broadcast:** the transmissions by a node is not only received by the designated receiver, but may also be overheard by near by nodes; **(2) interference:** transmissions from one node interfere with the transmission and reception capabilities of other nodes.

We tackle the Byzantine agreement in wireless networks by first focusing on the interference aspect of the wireless medium. We consider a n-node single-hop-single-channel system consists of one source and $n-1$ peers, in which all links have the same capacity C. For interference model, we assume that at most one node can transmit at a time, otherwise the transmissions collide and no data can be decoded by any node in the network. We assume that all communication channels/links are private such that only the designated receiver is able to retrieve the information from a successful transmission. This can be achieved in spite of the broadcast nature of the wireless medium by assigning different cryptographic keys to every pairs of nodes, and having the transmitter encrypt the out-going data with the key shared with the designated receiver. Otherwise we make no cryptographic assumptions. We assume a centralized controller that schedules transmissions in the networks, and every node must follow the assigned schedule. In terms of the adversary, we assume that the adversary has complete knowledge on the Byzantine agreement algorithm and the information being sent by every node. The adversary can take over nodes at any point during the algorithm, up to the point of taking over up to a $t < n/3$ nodes, including the source. The compromised nodes must follow the schedule decided by the centralized controller, but can engage in any other kind of deviations from the algorithm, including false messages and collusion. Given these assumptions above:

- We show that there exists an algorithm which computes Byzantine agreement deterministically on an l-bit message in a network with n nodes and at most $t < n/3$ faulty nodes, and uses $\frac{n(n-1)}{n-t}l + l^{1/2}O(n^4)$ bits of communication.

If we let l approach infinity and consider the average number of bits used for agreeing on 1 bit, then we have the following result on the sum capacity

- For the network with n nodes and at most $t < n/3$ faulty nodes to achieve agreement throughput of R bits/unit time, it is sufficient to have sum capacity $C > \frac{n(n-1)}{n-t}R$.

3.1 Sketch of the Algorithm

The algorithm here is similar to the one we presented in Section 2.1. In round 1, the source node divides $(n-t)c$ information bits into $n-t$ packets of size c bits, each packet being a symbol from GF(2^c). The source node encodes the $n-t$ packets of data into $2(n-1)$ packets, each of which is obtained as a linear combination of the $n-t$ packets of data. Then the source node sends 2 coded packets to each peer. In round 2, each peer sends the first packet received from the source to every other peer. It is to be noted that by the end of round 2, every fault-free peer has received n coded packets. Similar to the algorithm in Section 2.1, we can show that if the faulty nodes misbehave, at least one fault-free peer will detect the failure.

If a failure is detected, the broadcast phase is carried out in the same way as the algorithm in Section 2.1. Let us call a pair of nodes (i, j) is marked as f if the fault-free nodes can be sure that at least one of two nodes i, j is faulty. We can show that after the broadcast phase, at least one pair of nodes will be marked as f. In the subsequent generations, the schedule is modified slightly such that there is no transmission on the links between nodes i and j if the pair (i, j) is marked as f, and any further misbehavior will be detected. Every time a failure is detected, at least one more pair of nodes will be marked as f after the corresponding broadcast phase. Since a fault-free node can appear in at most t pairs marked as f, all faulty nodes will be identified with at most $t(t+1)$ broadcast phases.

By a suitable choice of c, we are able to upper bound the total number of bits communicated to achieve agreement of an l-bit message by $\frac{n(n-1)}{n-t}l + l^{1/2}O(n^4)$. Hence, agreement throughput of R bits/unit time can be achieved with the sum capacity of the system C to be arbitrarily close to $\frac{n(n-1)}{n-t}R$ from above, by choosing a large enough l. The detailed description of the algorithm and analysis can be found in our recent technical report [4].

4. DISCUSSION

In this paper, we have investigated two extreme cases: wired networks with point-to-point links, and single-hop wireless networks in which all links share the sum capacity. We are currently working on solving Byzantine agreement under more general model of the wireless medium.

In our other works [5, 6], we have explored the benefits of the broadcast nature of the wireless medium for reliable unicast/multicast in networks subject to Byzantine node failures.

5. REFERENCES

[1] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4:382–401, 1982.

[2] G. Liang and N. Vaidya. Capacity of Byzantine Agreement: Complete Characterization of the Four Node Network. *Technical Report, CSL, UIUC*, April 2010.

[3] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[4] G. Liang and N. Vaidya. Complexity of Multi-Value Byzantine Agreement. *Technical Report, CSL, UIUC*, June 2010.

[5] G. Liang, R. Agarwal, and N. Vaidya. When Watchdog Meets Coding. *INFOCOM 2010. 29th IEEE International Conference on Computer Communications*, March 2010.

[6] G. Liang, R. Agarwal, and N. Vaidya. Secure Capacity of Wireless Broadcast Networks. *Technical Report, CSL, UIUC*, September 2009.