BIT OPTIMAL DISTRIBUTED CONSENSUS[1]

Piotr Berman,[1] Juan A. Garay,[2] and Kenneth J. Perry[2]

[1] Department of Computer Science
The Pennsylvania State University
University Park, PA 16802

[2] I.B.M. T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, New York 10598

ABSTRACT

The *Distributed Consensus* problem involves n processors each of which holds an initial binary value. At most t processors may be faulty and ignore any protocol (even behaving maliciously), yet it is required that non-faulty processors eventually agree on a common value that was initially held by one of them. The quality of a consensus protocol is measured using the following parameters: the number of processors n, the number of rounds of message exchange r and the total number of bits transmitted B. The known lower bounds are respectively $3t+1$, $t+1$ and $\Omega(nt)$.

While no known protocol is optimal in all three aspects simultaneously, the protocol presented in this paper takes further steps in this direction: it achieves for the first time asymptotically optimal bit transfer ($B = O(nt)$), together with optimal number of processors n and nearly optimal r. Previously existing consensus protocols required $B = \Omega(n^2 t)$, regardless of the other parameters' values.

INTRODUCTION AND PROBLEM STATEMENT

The *Distributed Consensus* problem is one of the central topics in the theory of reliable distributed systems. Given a set P of processors, we are told that some unknown subset T of size no more than t are faulty. The goal of the Consensus problem is to cause the *correct* processors (i.e., those in $P - T$) to agree on an outcome regardless of

[1]Appears in *Computer Science Research* (eds. R. Yaeza-Bates and U. Manber), Plenum Publishing Corporation, NY, NY, pp. 313-322, 1992.

the behavior of the *faulty* processors (i.e., those in T). Motivated by practical concerns, a recent focus of research has been on the efficiency of protocols that solve the problem.

The standard model used in studying this problem is of a network of n synchronous processors numbered from 1 to n. The computation performed by the network evolves as a series of *rounds*.[2] In each round, the correct processors exchange messages over reliable network links and perform local computations according to the protocol. In contrast, the faulty processors may exhibit arbitrary (*Byzantine*) behavior, even appearing to act in concert maliciously. Each processor p starts with a binary value stored in local variable V_p. A protocol solves the binary Distributed Consensus problem if it always terminates and satisfies the following two conditions:

- *Agreement*: at termination $V_p = V_q$ for all $p, q \in P - T$.
- *Validity*: if $V_p = v$ for every $p \in P - T$ is true at the beginning of the computation, then it is also true at the end.

This is a variant of the *Byzantine Agreement* problem in which the processors are required to agree on a value sent to them by one of the processors (called the *general*) in an initial round [LSP].

There are several parameters that characterize the quality of a protocol that solves this problem. These parameters (with their optimal values indicated in parentheses) are:

- the number of processors n as a function of the maximum number of faulty processors t allowed ($n \geq 3t + 1$ [LSP]);
- the number of communication rounds r ($r \geq t + 1$ [DS]);
- the communication costs, alternatively given by the maximum message size m ($m \geq 1$, obvious), or the total number of transmitted bits B ($B \geq (n - t)(t + 1)/4$ [DR]).

Convinced that communication costs are one of the dominant factors in a protocol's performance [G], in this paper we focus on optimizing these costs, in particular B, the total bit transfer. Recent work by Hadzilacos and Halpern [HH] also addresses this concern by studying the message and bit complexity of runs that are failure-free.

All previous protocols (e.g., [BD, BDDS, BG1, BGP, LSP, MW]) require $B = \Omega(n^2 t)$, regardless of the value of the other parameters. The contribution of this paper is the first consensus protocol that achieves asymptotically optimal bit transfer (i.e., $B = O(nt)$) using the optimal number of processors ($n > 3t$) and nearly optimal number of rounds ($r = t + o(t)$)). A similar result has been concurrently and independently obtained by Coan and Welch [CW1].

The remainder of the paper is organized as follows. The first section describes the *Phase King* protocol of [BGP], a consensus protocol that is optimal in n and asymptotically optimal in r and m. The following section shows that the number of rounds required to run *Phase King* may be reduced by applying the *committee* technique of [BG1]. In the subsequent section, we further transform *Phase King*, resulting in a protocol that is optimal in total bit transfer. The resulting protocol uses the committee technique and consists of just two phases; in each phase the committee recursively runs

[2] A certain level of synchrony is necessary to obtain distributed consensus (see [FLP]). Thus every known protocol either assumes perfect synchrony, as here, or at least bounds the number of steps that a correct processor may execute before receiving a reply from another correct processor. Interesting new results in this model were obtained recently by Attiya *et al.* [ADLS].

```
for k := 1 to t + 1 do begin
                (* universal exchange 1 *)
    send(V)
    for j := 0 to 1 do
                C[j] := the number of received j's;

                (* universal exchange 2 *)
    for j := 0 to 1 do
                send(C[j] ≥ n − t)
                D[j] := the number of received 1's;
    V := D[1] > t;

                (* king's broadcast *)
    if k = p then send(V);
    if D[V] < n − t then
                V := the message received from k;
end;
```

Figure 1: Protocol *Phase King*, code for processor p.

the same protocol. We conclude with a discussion of recent developments and open problems.

CONSTANT MESSAGE SIZE WITH OPTIMAL RESILIENCY

The *Phase King* protocol of [BGP], shown in Figure 1, is asymptotically optimal with respect to r and m and is exactly optimal in the number of processors (i.e., $n > 3t$). The protocol uses messages of constant size (2 bits) and runs in $t+1$ phases, each consisting of three exchange rounds: two universal exchanges and the king's broadcast. A universal exchange ensures that a processor will hold a value in local variable V only if enough processors claim to hold that value; the arrays C and D record the number of processors claiming to hold each value prior to the two universal exchanges. During the king's broadcast, the processor designated as king of the phase broadcasts its V and each processor replaces its own value of V with the king's unless it is certain that all correct processors computed the same majority. We will show that a processor can conclude the latter if its own majority exceeds the required threshold by t. The protocol gets its name since a different processor is designated as king in each phase.

Theorem 1: Phase King *solves the Distributed Consensus problem using 3(t+1) exchange rounds and two-bit messages (or* $4(t+1)$ *rounds and single-bit messages), provided* $n > 3t$.

Proof: Observe that the second universal exchange can be implemented either as a single round with two-bit messages or two rounds with single-bit messages. A simple inspection of the code confirms the respective round and message size complexities obtained for each alternative. Correctness will be established using the following preliminary result. Throughout the paper X_p denotes the value of local variable X for processor p.

Persistency. If for all $p \in P - T$, $V_p = v$ is true at the start of a phase, it remains true at the end of the phase.

To see this, observe that after the first universal exchange each correct processor p has $C[v]_p \geq n-t > 2t$ and $C[\bar{v}]_p \leq t$. Consequently, in the second universal exchange, each correct processor sends 1 for v and 0 for $\bar{v}$, resulting in each correct processor p ignoring the king's broadcast since $D[v]_p \geq n-t$ and $D[\bar{v}]_p \leq t$. Thus $V_p = v$ for each correct processor p.

Now we can address the correctness of the protocol.
Agreement. Since the number of phases $(t+1)$ exceeds the number of faulty processors (t), there is some first phase g with a non-faulty king. At the end of phase g one of the following cases holds:

1. $D[V]_p < n-t$ for each $p \in P-T$. Then every correct processor assigns V_g to V during king g's broadcast.

2. $D[V]_p \geq n-t$ for some $p \in P-T$. Note that since there are no more than t faulty processors, $|C[b]_q - C[b]_s| \leq t$ and $|D[b]_q - D[b]_s| \leq t$, for every $q, s \in P-T$ and $b \in \{0,1\}$. Therefore if some correct processor q sends 1 for b in the second universal exchange, then no correct processor s can send 1 for $\bar{b}$ (since $C[b]_q \geq n-t$ implies $C[b]_s \geq n-2t$, $C[\bar{b}]_s$ must be at most $2t < n-t$). Let $V_p = v$. By the same logic, $D[V]_p \geq n-t$ implies that phase g's king (g) broadcasts v, since $D[V]_g \geq n-2t > t$. Therefore each correct processor q sets V to v, either because $D[V]_q \geq n-t$ or by receiving King g's broadcast.

Persistency assures that no correct processor subsequently alters V.
Validity. Follows trivially from Persistency. ∎

REDUCING THE NUMBER OF ROUNDS WITH COMMITTEES

While the consensus protocol presented in the previous section is processor-optimal and has a very small message size (1 or 2 bits), its round complexity is at least three times optimal. In this section we show how the committee technique of Berman and Garay [BG1] can be used to improve the round complexity to $t(1+1/d)$ rounds for every constant d, while maintaining the same number of processors and messages of constant size.

The idea is as follows. Correctness of the *Phase King* protocol depends on the existence of a phase in which the king is non-faulty; running for $(t+1)$ phases assures this. We can instead partition the processors into R disjoint committees, replace the king of phase k by committee Q_k, and run for R phases. By carefully choosing R, the committee size, and the intra-committee protocol, we can assure that the committees are valid substitutes for the phase kings, that a correct committee will occur, and that R is small. The tradeoff is that each phase requires multiple rounds for the intra-committee protocol to run so that the total number of phases is some constant times R.

To implement this idea, we modify the *Phase King* protocol to that shown in Figure 2. We change the number of phases in Fig. 1 from $t+1$ to R (where R will be established later) and replace the king's code by committee Q_k's code. Intra-committee the processors run protocol DCP, where DCP is any round-optimal distributed consensus protocol that satisfies the following properties:

1. in each round a correct processor sends the same message to all other processors;

```
for k := 1 to R do begin
                (* universal exchange 1 *)
    send(V);
    for j := 0 to 1 do
                C[j] := the number of received j's;

                (* universal exchange 2 *)
    for j := 0 to 1 do
                send(C[j] ≥ n − t)
                D[j] := the number of received 1's;
    V := D[1] > t;

                (* committee's broadcast *)
    if p ∈ Q_k then
                run DCP within Q_k using V as the initial
                value and sending the messages required by this
                protocol to all processors;
    if D[V] < n − t then
                V := the consensus value of Q_k;
end;
```

Figure 2: Reducing the round complexity of *Phase King*, code for processor p.

2. a recipient of these messages that is not a committee member can compute the consensus value; and

3. fewer than 1/3 of the processors may be faulty.

All published deterministic round- and processor-optimal protocols satisfy these properties [BDDS, BG2, LSP]. A round-optimal protocol is used intra-committee to accelerate consensus yet messages of only moderate size result because committees contain few processors.

Correctness of this idea will follow from *Phase King*'s correctness once we show that R committees can be chosen so as to guarantee the existence of one in which correct processors dominate. Let t_k denote the maximum number of faulty processors that can be tolerated by the DCP of the kth committee Q_k. By choosing an R that satisfies $\sum_{k=1}^{R}(t_k+1) = t+1$, it is possible to partition the n processors into R committees such that $|Q_k| = 3t_k+1$ since

$$\sum_{k=1}^{R}(3t_k+1) = 3\sum_{k=1}^{R}(t_k+1) - 2R = 3(t+1) - 2R = 3t+1-2(R-1) \leq n.$$

By the pigeonhole principle there is at least one committee Q_k with no more than t_k faulty members. The DCP run by this Q_k succeeds in broadcasting the correct members' consensus to all correct processors and thus the members collectively serve the role of the non-faulty king required in *Phase King*'s correctness proof.

Now we can analyze the number of rounds of the resulting protocol. Phase k requires two universal exchanges plus the number of rounds necessary for the DCP Q_k, which is t_k+1 since DCP is round optimal. Thus, the entire protocol uses $r = \sum_{k=1}^{R}(t_k+1+2) = t+1+2R$ rounds. This allows us to prove

Theorem 2: *For each $d > 1$ there exist a constant m_0 and a Distributed Consensus protocol that uses $n > 3t$ processors, $r \leq t(1 + 1/d) + 3$ rounds and messages of size m_0.*

Proof: The required protocol is an instance of the protocol in Figure 2. Our choice for DCP is any processor- and round-optimal consensus protocol satisfying the three conditions specified above (e.g. [BG2]); we simply argue that for a bounded committee size any DCP will yield bounded message size. We need to assure $t_k < 2d$, which is achieved as follows: choose $R = \lceil (t+1)/2d \rceil$, $t_k = 2d - 1$ for $1 \leq k \leq (R-1)/d$, and $t_R = [(t+1) \bmod 2d] - 1$. The resulting number of rounds is $t + 1 + 2R = t + 1 + 2\lceil (t+1)/2d \rceil < t(1 + 1/d) + 3$. ∎

Remark: The size of the constant in Theorem 2 depends on the choice of a protocol which is optimal in both r and n. The classic protocol of Lamport *et al.* yields $m_0 = d^{O(d)}$ [LSP]. In [BG2] Berman and Garay describe protocol *ESDM* with messages of size $O(1.5^t)$, yielding $m_0 = O(1.5^d)$. Admittedly, the existing round- and processor-optimal protocols make Theorem 2 practical only for small values of d.

CONSENSUS WITH OPTIMAL BIT TRANSFER

In this section we show that it is possible to reach consensus with asymptotically optimal bit transfer (i.e. $B = O(nt)$). The resulting protocol also achieves processor optimality and near round optimality. In order to obtain this result we apply the method of the previous section in the extreme: the processors are split into two committees, each acting the role of king in one of two phases of the *Phase King* protocol. Intra-committee, the processors reach consensus by recursively applying the technique until the committee size becomes small, in which case they use the *ESDM* protocol of [BG2]. This results in only 6 exchange rounds involving all n processors, 12 rounds involving about $n/2$ processors, 24 rounds involving $n/4$ processors, etc. Thus, the total bit transfer is $cn^2(1 + 1/2 + 1/4 + ...)$. The choice of *ESDM* as the basis results in near round optimality, as we shall demonstrate.

To describe the technique in detail, let Q_1 be equal to P, the set of all processors. Q_w is recursively partitioned into two committees Q_{2w} and Q_{2w+1} of balanced size. We call the resulting protocol *Recursive Phase King* (see Figure 3). At the top level, the processors execute $RPK(1,V)$, where V contains the processor's initial binary value. The value of parameter M, which appears in the protocol, will be computed later.

Theorem 3: Recursive Phase King *solves the Distributed Consensus problem using optimal number of processors ($n > 3t$), $r = t + o(t)$ communication rounds, and total bit transfer $B = O(nt)$.*

Proof: *Correctness.* If $|Q_w| > 3|Q_w \cap T|$ then in the call to $RPK(w, V)$ the values of V of the processors from $Q_w - T$ satisfy the *Validity* and *Agreement* conditions. In the case when $|Q_w| \leq M$, the claim follows from the correctness of $ESDM$. In the other case, note that either $|Q_{2w}| > 3 \cdot |Q_{2w} \cap T|$, or $|Q_{2w+1}| > 3 \cdot |Q_{2w+1} \cap T|$. The subcommittee with the favorable ratio plays the role of the honest king in *Phase King*, and its proof of correctness applies (Theorem 1).
Number of rounds. The number of rounds required by RPK can be expressed as the following recurrence (where $|Q_w| = s$):

```
procedure RPK(w: integer; var V: {0,1});
var C, D: array[0..2] of integer;
    U: {0,1}; t_w: integer;
begin
    t_w := (|Q_w| − 1) div 3;
    if |Q_w| ≤ M then
            ESDM(w, V)
    else for k := 0 to 1 do begin
            perform universal exchanges of Phase King within Q_w,
            which causes arrays C, D and possibly variable V to be set;
            if p ∈ Q_{2w+k} then begin
                  U := V;
                  RPK(2w + k, U);
                  send U to Q_{2w+(1−k)}
            end;
            if p ∈ Q_{2w+(1−k)} then
                  U := the most frequent value received
                  from members of Q_{2w+k};
            if D[V] < |Q_w| − t_w then
                  V := U;
    end
end;
```

Figure 3: *Recursive Phase King*, code for processor p.

$$r(s) = \begin{cases} \lfloor s/3 \rfloor + 1 & \text{if } s \le M \\ r(\lfloor s/2 \rfloor) + r(\lceil s/2 \rceil) + 6 & \text{otherwise} \end{cases}$$

For example, for $s = 2^k M$ processors this recurrence yields $r(s) = (2^k - 1)6 + 2^k(\lfloor M/3 \rfloor + 1) < 2^k(\lfloor M/3 \rfloor + 7) \le \frac{s}{3}(1 + \frac{21}{M})$. In general, $r(3t+1) = t + o(t)$, provided $M = \omega(1)$.
Total bit transfer. From Fig. 3, the total number of bits communicated can be expressed by the recurrence:

$$B(s) \;=\; \begin{cases} c2^s & \text{if } s \le M \\ B(\lfloor s/2 \rfloor) + B(\lceil s/2 \rceil) + 6.5s^2 & \text{otherwise} \end{cases}$$

For $s = n = 2^k M$ processors this recurrence yields $6.5M^2(2^{2k} + 2^{2k-1} + \cdots + 2^k) + c2^k 2^M \le 13(2^k M)^2 + 2^k M \frac{c2^M}{M} \le 14(2^k M)^2 = 14n^2$, provided $\frac{c2^M}{M} \le n$.

The latter is true for $M < \alpha \log n$, where α is a suitable constant. ∎

CONCLUSIONS AND OPEN PROBLEMS

In this paper we have presented a simple Distributed Consensus protocol that achieves asymptotically optimal bit transfer ($B = O(nt)$), together with the optimal number of processors and nearly optimal number of communication rounds. Previous consensus protocols required $\Omega(n^2 t)$ bit transfer, regardless of the other parameters' values. The same result was independently and concurrently obtained by Coan and Welch [CW1].

The starting building block for our result is the *Phase King* protocol of [BGP]. This simple protocol has proven useful in other directions as well: An "early-stopping" version of the protocol has been recently used by Bar-Noy *et al.* [BDGK] to achieve optimal consensus in the *amortized* sense. Namely, they look at the cost of consensus when it is repeatedly needed, showing in particular that consensus is possible in *constant* amortized number of rounds.

In an interesting new development, Coan and Welch show how to achieve consensus with basically the same parameters as in this paper, but with 1-bit messages [CW2]. Finally, it remains open whether Distributed Consensus is achievable in the optimal number of rounds, using the optimal number of processors and polynomial communication. The best known algorithm so far is that of [BG2], which achieves consensus in the optimal number of rounds, with $n = (3 + \varepsilon)t$ processors and messages of size $O(2^{1/\varepsilon}\text{poly}(t))$ (i.e., polynomial communication for $\varepsilon \geq 1/\log t$).

References

[ADLS] H. Attiya, C. Dwork, N. Lynch and L. Stockmeyer, "Bounds on the time to reach agreement in the presence of timing uncertainty," *Proc. 23rd STOC*, pp. 359-369, May 1991.

[BD] A. Bar-Noy and D. Dolev, "Families of Consensus Algorithms," *Proc. 3rd Aegean Workshop on Computing*, pp. 380-390, June/July 1988.

[BDDS] A. Bar-Noy, D. Dolev, C. Dwork and H.R. Strong, "Shifting gears: changing algorithms on the fly to expedite Byzantine Agreement," *Proc. 6th PODC*, pp. 42-51, August 1987.

[BDGK] A. Bar-Noy, X. Deng, J. Garay and T. Kameda, "Optimal Amortized Distributed Consensus," to appear in *Proc. 5th International Workshop on Distributed Algorithms* (*LNCS*), Delphi, Greece, October 1991.

[BG1] P. Berman and J.A. Garay, "Asymptotically Optimal Distributed Consensus," *Proc. ICALP 89*, LNCS Vol. 372, pp. 80-94, July 1989.

[BG2] P. Berman and J.A. Garay, "Efficient Distributed Consensus with $n = (3+\varepsilon)t$ Processors," to appear in *Proc. 5th International Workshop on Distributed Algorithms* (*LNCS*), Delphi, Greece, October 1991.

[BGP] P. Berman, J.A. Garay and K.J. Perry, "Towards Optimal Distributed Consensus," *Proc. 30th FOCS*, pp. 410-415, October/November 1989.

[CW1] B.A. Coan and J.L. Welch, "Modular Construction of a Byzantine Agreement Protocol with Optimal Message Bit Complexity," *Proc. 27th Allerton Conference on Communication, Control, and Computing*, pp. 1062-1071, 1989. To appear in *Information and Computation.*

[CW2] B.A. Coan and J.L. Welch, "Modular Construction of an Efficient 1-Bit Byzantine Agreement Protocol," personal communication. To appear in *Mathematical Systems Theory*, special issue dedicated to Distributed Agreement.

[DR] D. Dolev and R. Reischuk, "Bounds of Information Exchange for Byzantine Agreement," *JACM*, Vol. 32, No. 1, pp. 191-204, 1985.

[DS] D. Dolev and H.R. Strong, "Polynomial Algorithms for Multiple Processor Agreement," *Proc. 14th STOC*, pp. 401-407, May 1982.

[FLP] M. Fisher, N. Lynch and M. Paterson, "Impossibility of Distributed Consensus with one faulty process," *JACM*, Vol. 32, No. 2 (1985), pp. 374-382.

[G] J. Gray, "The Cost of Messages," *Proc. 7th PODC*, pp. 1-7, August 1988.

[HH] V. Hadzilacos and J. Halpern, "Message-Optimal Protocols for Byzantine Agreement," *Proc. 10th PODC*, pp. 309-324, August 1991..

[LSP] L. Lamport, R.E. Shostak and M. Pease, "The Byzantine Generals Problem," *ACM ToPLaS*, Vol. 4, No. 3, pp. 382-401, July 1982.

[MW] Y. Moses and O. Waarts, "Coordinated Traversal: $(t+1)$-Round Byzantine Agreement in Polynomial Time," *Proc. 29th FOCS*, pp. 246-255, October 1988.