ASYMPTOTICALLY OPTIMAL DISTRIBUTED CONSENSUS *

Piotr Berman† *Juan A. Garay*‡ *Kenneth J. Perry*‡

ABSTRACT

The *Distributed Consensus* problem requires correct processors to reach consensus in the presence of the arbitrary behavior of t faulty processors. The quality of a protocol that solves this problem is measured by the following parameters: total number of processors n, number of communication rounds r, and communication complexity, alternatively given by the maximal message size m, or the total number of bits transmitted B. The known lower bounds are respectively $3t+1$, $t+1$, 1 and $n(t+1)/4$.

In this paper we present two protocols that for the first time achieve consensus with the cost parameters—n, r and m—within a constant factor of their optima. Both protocols use the "phase king" approach in which processors take charge of the computation for one round. The first protocol achieves consensus with $n=4t+1$, $r=2(t+1)$ and $m=1$. The second protocol uses $n=3t+1$, $r=3(t+1)$ and $m=2$.

By replacing "phase kings" by committees of processors we subsequently present two families of protocols. These families are indexed by d and have complexities $r \sim t(1+1/d)$, $m=O(d^5)$ and $r \sim t(1+1/d)$, $m=O(d^3 4.33^d)$, respectively, while using the same number of processors as the original protocols.

Finally, we apply the committee technique recursively to obtain consensus with $n=3t+1$, $r=t+o(t)$ and $B=O(nt)$. All our algorithms have succint descriptions and work in polynomial time.

---

* Parts of this paper presented in *ICALP 89, FOCS 89* and *WDAG-4*. This work was partially supported by AFOSR contract 87-0400 and NSF grant CR 8805978.

† Department of Computer Science, The Pennsylvania State University, University Park, PA 16802.

‡ IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598.

1. Introduction

The *Distributed Consensus* problem is one of the central topics in the theory of reliable distributed systems. We are given a set of processors P of which some unknown subset T are not guaranteed to follow their protocol. Consensus is achieved if the *good* processors (those in $P-T$) agree on an outcome regardless of the behavior of the *bad* processors (those in T). A focus of recent research has been the efficiency of algorithms that solve the problem. This aspect is essential in practical settings.

We use the standard model of a synchronous network of n processors numbered $\{1, \ldots, n\}$. The computation evolves as a series of *rounds* during which the good processors receive and send messages over the network links and perform local computations according to the protocol. We assume that the links are reliable. The bad processors may exhibit arbitrary (*Byzantine*) behavior, even appearing to act maliciously in concert.

We assume that each processor starts with an initial binary value stored in local variable V. X_p denotes the value of X for processor 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 at the beginning of the computation $V_p = v$ for every $p \in P-T$, then the same holds 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 basic aspects contributing to the quality of an algorithm for Distributed Consensus; here we consider the total number of processors n as a function of the maximum allowed number of faulty processors t, the number of rounds r, and the communication costs. One measure of the latter is the maximum bit size m of a single message (as in [BD]); another measure considered in the literature is the total number B of bit transmissions. (Independently, [CW] also provides the best algorithm so far with respect to B.)

The lower bounds for n, r, m and B are respectively $3t+1$ [LSP], $t+1$ [DS], 1 and $n(t+1)/4$ [DR]. No protocol to date has been proven to be optimal in B. Protocols that are optimal in any two of the other aspects are known, but no protocol optimal in all three has been discovered yet. Known protocols optimal in n and r have exponential m [BDDS, LSP].

There are many protocols having optimal n and polynomial communication costs, but $r = O(t)$ [BDDS, C1, DS, TPS]. Coan presented a family of agreement protocols with $n > 4t$ that use $t + t/d$ rounds and messages of size $O(n^d)$, for every d, thus making the tradeoff between number of rounds and message size explicit [C1]. (These algorithms, however, required exponential amount of local computation; this was improved to polynomial by Bar-Noy *et al.* [BDDS].) Toueg *et al.* achieved optimal n, $r = O(t)$ and $m = \Omega(\log n)$ (but $B = O(nt \log n)$ [TPS]. Recent algorithms reduce the communication costs even further [CW, this paper], while remaining optimal in n. In particular, Coan and Welch showed how to simultaneously achieve $B = O(nt)$ and $r = t + o(t)$. The current paper shows a series of protocols sharing a common principle, which also obtains this combination of parameters.

The reduction of communication costs appears to be much more difficult when one must achieve optimality in the number of rounds. Bar-Noy and Dolev [BD] and Coan [C2] showed how to achieve small message size and optimal r with $n = O(t^2)$ (in particular, the *Beep Once* algorithm of [BD] uses single-bit messages). The round-optimal algorithm of Moses and Waarts [MW] requires $n > 6t$ and uses messages of polynomial size. In [BGP] we have improved the processor requirement to $n > 4t$.

On the other hand, even assuming that each communication round has a large cost, in any practical setting the message size will quickly dominate the cost if it is proportional to, say, n^3. This point makes the protocols with small communication costs important even if they use sub-optimal number of rounds.

In Section 2 of this paper, we present two simple algorithms, the first with $n > 4t$, $r = 2(t+1)$ and $m = 1$, the second with $n > 3t$, $r = 3(t+1)$ and $m = 2$ (or $r = 4(t+1)$ and $m = 1$). These are the first protocols simultaneously having n, t and m within a constant factor of their optima. The total bit transfer of these protocols is of the order nt^2. Section 3 shows how to decrease the bit transfer to asymptotically optimal

without increasing the message size. Section 4 shows how to reduce the rounds to near optimal ($r = t + o(t)$) while only moderately increasing the message size and increasing B by a small constant factor. All of our protocols have local computations of polynomial time.

In our protocols we use send(X) to denote sending the value of X to all processors; we also use $\bar{v}$ to denote $1-v$.

2. Simple Protocols with Constant Message Size

```
for k := 1 to t+1 do begin
        (* universal exchange *)
    send(V)
    for j := 0 to 1 do
        C[j] := the number of received j's;
    V := C[1] > 2t;
        (* queen's broadcast *)
    if k = p then send(V);
    if C[V] < n-t then
        V := the received message
end;
```

Fig. 1. Protocol *Phase Queen*, code for processor p.

The first consensus protocol—*Phase Queen*—which requires $n > 4t$ and uses single-bit messages is shown in Figure 1. It executes $t+1$ phases, each consisting of two communication rounds: the universal exchange and the broadcast of the "phase queen;" each phase has a distinct queen. We assume that each processor p has local variable V (the subject of the consensus problem) and integer array $C[0..1]$ (the counts of 0's and 1's received). In the universal exchange each processor changes V to what it perceives as the majority value. During the queen's broadcast a processor replaces this value with the Queen's message unless it is certain that all good 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.

Theorem 1: *Phase Queen* solves the Distributed Consensus problem using $2(t+1)$ exchange rounds and single-bit messages, provided $n > 4t$.

Proof: The number of rounds and the message size follow trivially from the code. As a preliminary step, we show the following property of the protocol.

Persistency. If for all $p \in P-T$, $V_p = v$ is true at the start of a phase, it remains true at the phase's end.

To see this, once the property becomes true the universal exchange results in every good processor p having $C[v]_p \geq n-t > 2t$ and $C[\bar{v}]_p \leq t < 2t$. Therefore it computes $V_p = v$ and disregards the Queen's broadcast.

Now we can address the correctness of the protocol.

Agreement. Since the number of phases $(t+1)$ exceeds the number of bad processors (t), there is some first phase g with a good Queen. One of the following cases holds at the end of phase g:

1) $C[V]_p < n-t$ for each $p \in P-T$. Then every correct processor assigns V_g to V during the Queen's broadcast of phase g.

2) $C[V]_p \geq n-t$ for some $p \in P-T$. Among the $n-t$ processors from which p received value v, at most t are bad so at least $n-2t$ are good. Thus, *every* good processor received v at least $n-2t > 2t$ times, and $\bar{v}$ at most $2t$ times. As a consequence, at the end of the universal exchange $V_q = v$ for every $q \in P-T$, including Queen g; therefore the Queen's broadcast must result in $V_q = v$ for all $q \in P-T$.

Persistency assures that no good processor changes its value subsequent to phase g.

Validity. Follows trivially from persistency. □

The second consensus protocol—*Phase King* —is shown in Figure 2. Like *Phase Queen*, this is an asymptotically optimal consensus protocol with respect to r, n and m, but it is also 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. Each processor has a local variable V, and integer arrays C and D (the counts of 0's and 1's received in two universal exchanges). The philosophy behind *Phase King* is the same as in *Phase Queen*. However, correctness requires that if all good processors start a phase with some value v, then each deduces this fact and forever more resists the King's broadcast. Because the number of processors is smaller than in *Phase Queen*,

```
for k := 1 to t+1 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 begin
        send(C[j] > n-t);
        D[j] := the number of received 1's;
    end;
    V := D[1] > t;
                    (* King's broadcast *)
    if k = p then send(V);
    if D[V] < n-t then
        V := the received message;
end;
```

Fig. 2. The *Phase King* protocol: code for processor i.

two universal exchanges are needed to achieve this.

Theorem 2: *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: The number of rounds and the message size follow trivially from the code. The second universal exchange can be implemented as a single round with two-bit messages or two rounds with single-bit messages. As in the previous proof, we initially show that Phase King satisfies

Persistency. If for all $p \in P-T$, $V_p = v$ is true at the start of a phase, it remains true at the phase's end.

After the first universal exchange each good processor p has $C[v]_p \geq n-t$ and $C[\bar{v}]_p \leq t < n-t$. Consequently, in the second universal exchange, each good processor sends 1 for v and 0 for $\bar{v}$, resulting in each good 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 good processor p.

Agreement. As in the proof of *Phase Queen*, let g be the first phase with a good king. One of the following cases holds at the end of phase g:

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 bad 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 good processor q sends 1 for b in the second universal exchange, then no good processor s can send 1 for $\bar{b}$ (since $C[b]_q \geq n-t$ implies $C[b]_s \geq n-2t$, $\overline{C[b]}_s$ 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 good 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 good processor subsequently alters V.

Validity. Follows trivially from persistency. □

3. Protocols with Reduced Number of Rounds

While the protocols presented in the previous section clearly have optimal message size, their round complexity is at least twice the optimal. Here we describe a method which allows us to convert *Phase Queen* to a protocol with constant message size and $t(1+1/d)$ rounds, for every constant d, while maintaining the same number of processors. The same method can also be applied to Phase King.

The idea is as follows. Correctness of the *Phase Queen* protocol depends on the existence of a phase in which the Queen is good; running for $(t+1)$ phases assures this. We can instead partition the processors in to R disjoint committees, replace the Queen 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 Queens, that a good 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 Queen* protocol to that shown in Figure 3. We change the number of phases in Fig.1 from $t+1$ to R (where R will be established later) and replace the Queen'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 (because the committees are small, they can internally apply a round-optimal protocol, thereby accelerating

the consensus, and yet use messages of moderate size): (a) in each round a good processor sends the same message to all other processors, (b) a recipient of these messages that is not a committee member can compute the consensus value, and (c) fewer than 1/4 of the processors may be bad. All published deterministic round-optimal protocols satisfy these properties [BDDS, BGP, LSP].

```
for k := 1 to t+1 do begin
        (* universal exchange *)
    send(V)
    for j := 0 to 1 do
        C[j] := the number of received j's;
    V := C[1] > 2t;
        (* 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 C[V] < n−t then
        V := the consensus value of Q_k
end;
```

Fig. 3. Reducing the round complexity of *Phase Queen*.

Correctness of this idea will follow from *Phase Queen*'s correctness once we show that R committees can be chosen so as to guarantee the existence of one in which good processors dominate. Let t_k denote the maximum number of bad processors that may be overcome 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 $|Q_k| = 4t_k+1$ since

$$\sum_{k=1}^{R}(4t_k+1) = 4\sum_{k=1}^{R}(t_k+1) - 3R = 4(t+1)-3R = 4t+1-3(R-1) \le n \ .$$

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

Now we can analyze the number of rounds of the resulting protocol. Phase k requires one universal exchange 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 $r = \sum_{k=1}^{R} (t_k+1+1) = t+1+R$ rounds.

Theorem 3: For every $d>1$ there exists a Distributed Consensus protocol that uses $n>4t$ processors, $r \le t(1+1/d)+2$ rounds and messages of size $m = O(d^5)$.

Proof: The required protocol is an instance of the protocol in Figure 3. Our choice for the round-optimal *DCP* is *Cloture Votes* [BGP]. Because *Cloture Votes* has messages of size $O(t^5)$, it is necessary that the t_k's be smaller than d. We achieve this as follows: choose $R = \lceil (t+1)/d \rceil$, $t_k = d-1$ for $1 \le k \le (R-1)/d$, and $t_R = [(t+1) \bmod d\,]-1$. This choice satisfies $\sum_{k=1}^{R} (t_k+1) = t+1$ as required. The resulting number of rounds is $t+1+R = t+1+\lceil (t+1)/d \rceil \le t(1+1/d)+2$. □

The committee technique can also be applied to the *Phase King* protocol with only minor modifications. Again the number of phases is reduced to R and R disjoint committees are formed, except that now $|Q_k| = 3t_k+1$ (the t_k's again must satisfy $\sum_{k=1}^{R} (t_k+1) = t+1$). As before, we can verify that this condition assures the formation of committees such that at least one Q_k has no more than t_k bad members. The number of rounds in the kth phase is now $2+t_k+1$, and so the total number of rounds equals $t+1+2R$. This allow us to prove

Theorem 4: For each $d>1$ there exist a constant m_o and a Distributed Consensus protocol that uses $n>3t$ processors, $r \le t(1+1/d)+3$ rounds and messages of size m_o.

Proof: The proof follows the same reasoning as in Theorem 3 with the following exceptions. Instead of applying *Cloture Votes* we simply argue that for a bounded committee size any *DCP* will yield bounded message size. Here we need to assure $t_k < 2d$, which is achieved as follows: choose $R = \lceil (t+1)/2d \rceil$, $t_k = 2d-1$ for $1 \le k \le (R-1)/d$, 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 4 depends on the choice of a protocol which is optimal in both r and n. The classic protocol of Lamport *et al.* yields $m_o = d^{O(d)}$ [LSP]. In [BG2] Berman and Garay describe protocol $ESFM_{3t}$ with messages of size $O(t^3 2.08^t)$, yielding $m_o = O(d^3 4.33^d)$. Admittedly, the existing round- and processor-optimal protocols make Theorem 4 practical only for small values of d.

4. Consensus with Asymptotically Optimal Bit Transfer

In order to achieve asymptotically optimal bit transfer, 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 $ESFM_{3t}$ 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 $cn^2(1+1/2+1/4+\cdots)$. The choice of $ESFM_{3t}$ as the basis results in near round optimality,as we shall demonstrate.

To describe the technique in detail, let Q_1 be the set of all processors (P). Q_w is recursively partitioned into two committees Q_{2w} and Q_{2w+1} of balanced size. *Recursive Phase King* is given in Figure 4. 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 5: *Recursive Phase King* solves the Distributed Consensus problem using optimal number of processors ($n > 3t$), $t + o(t)$ communication rounds, and total bit transfer $O(nt)$.

Proof:

Correctness. If $|C_w| > 3|Q_w \cap T|$ then in the call to $RPK(w, V)$ the values of V of processors from $C_w - T$ satisfy the *Validity* and *Agreement* conditions. In the case when $|C_w| \le M$, the claim follows from the correctness of $ESFM_{3t}$. In the other case, note that either $|C_{2w} > 3|C_{2w} \cap T|$, or $|C_{2w+1} > 3|C_{2w+1} \cap T|$. The subcommittee with the favorable ratio plays the role of the honest king in *Phase King*, and its proof of correctness applies (Theorem 2).

```
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
        ESFM_3t(w,V)
    else for k := 0 to 1 do begin
        perform universal exchanges of Phase King within Q_w;
        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 frequently received value;
        if D[V] < |Q_w|– t_w then
            V := U;
    end
end;
```

Fig. 4. *Recursive Phase King*: code for processor p.

Number of rounds. The number of rounds required by *RPK* can be expressed as the following recurrence (where $|Q_w| = s$):

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

For example, for $n = 2^k M$ 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. 4, the number of bits communicated can be expressed by the recurrence:

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

For $n = 2^k M$ processors this recurrence yields $6.5M^2(2^{2k} + 2^{2k-1} + \cdots + 2^k) + c\,2^k 2^M \le 6.5(2^k M)^2 + 2^k M \frac{c\,2^M}{M} \le 10(2^k M)^2 = 10n^2$, provided $\frac{c\,2^M}{M} < 3.5n$.

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

5. Conclusions

We have presented two simple protocols that achieve consensus with all cost parameters—n, r and m—simultaneously within a constant factor from their optima. Another virtue of these protocols is that they can be adapted to run very efficiently on bounded-degree networks [BG1]. Subsequently, by replacing "phase kings" by committees of processors we presented equally-resilient number of rounds/message size tradeoffs that improve on previous results [BDDS, C1]. Finally, by applying the committee technique recursively, we showed how to achieve consensus with asymptotically optimal total bit transfer and nearly-optimal number of rounds ($t+o(t)$).

An interesting question is whether these protocols can be modified to satisfy the *early stopping* property [DRS]. We note that in both cases, even when a processor knows that value v is being adopted unanimously by the good processors before the ruler's broadcast, it does not know whether other processors will also resist the ruler's message—bad ruler can destroy momentary consensus.

Acknowledgements

We wish to thank Brian Coan and Georg Schnitger for helpful suggestions and discussions. The first two authors are also grateful to the Department of Computer Science of the University of Chicago for its hospitality during the final stages of the preparation of this manuscript.

References

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

[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.

[BG1] P. Berman and J.A. Garay, "Asymptotically Optimal Distributed Consensus," *Proc. ICALP 89*, LNCS Vol. 372, pp. 80-94, July 1989. March 1990.

[BG2] P. Berman and J.A. Garay, "Improved Protocols for Distributed Consensus," The Pennsylvania State University, CS Department Tech. Report CS-90-15, March 1990.

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

[C1] B. Coan, "A communication-efficient canonical form for faul-tolerant distributed protocols," *Proc. 5th PODC*, pp. 63-72, August 1986. Revised version in *Achieving Consensus in Fault-Tolerant Distributed Computer Systems: Protocols, Lower Bounds, and Simulations*, Ph.D. dissertation, Massachusetts Institute of Technology, June 1987.

[C2] B. Coan, "Efficient agreement using fault diagnosis," *Proc. 26th Annual Allerton Conference on Communication, Control and Computing*, pp. 663-672, 1988.

[CW] B. Coan and J. Welch, "A Byzantine Agreement Protocol with Optimal Message Bit Complexity," *Proc. 27th Annual Allerton Conference on Communication, Control and Computing*, pp. 295-306, 1989.

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

[DRS] D. Dolev, R. Reischuk and H.R. Strong, "Early Stopping in Byzantine Agreement," IBM Research Report *RJ 5406(55357)*, 1986. To appear in *JACM*.

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

[FM] P. Feldman and S. Micali, "Optimal Algorithms for Byzantine Agreement," *Proc. 20th STOC*, pp. 148-161, May 1988.

[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.