Knowledge and Common Knowledge in a Byzantine Environment I: Crash failures

(Extended Abstract)

Cynthia Dwork
IBM Almaden Research Center,
San Jose, CA 95193

Yoram Moses
MIT Laboratory for Computer Science,
Cambridge, MA 02139

ABSTRACT

By analyzing the states of knowledge that the processors attain in an unreliable system of a simple type, we capture some of the basic underlying structure of such systems. The analysis provides us with a better understanding of existing protocols for problems such as Byzantine agreement, generalizes them considerably, and facilitates the design of improved protocols for many related problems.

1. Introduction

The problem of designing effective protocols for distributed systems whose components are unreliable is both important and difficult. In general, a protocol for a distributed system in which all components are liable to fail cannot unconditionally guarantee to achieve non-trivial goals. In particular, if all processors in the system fail at an early stage of an execution of the protocol, then fairly little will be achieved regardless of what actions the protocol intended for the processors to perform. However, such universal failures are not very common in practice, and we are often faced with the problem of seeking protocols that will function correctly so long as the number, type, and pattern of failures during the execution of the protocol are reasonably limited. A requirement that is often made of such protocols is *t-resiliency* — that they be guaranteed to achieve a particular goal so long as no more than t processors fail.

A good example of a desirable goal for a protocol in an unreliable system is called *Simultaneous Byzantine Agreement* (SBA), a variant of the Byzantine agreement problem introduced in [PSL]:

> Given are n processors, at most t of which might be faulty. Each processor p_i has an initial value $x_i \in \{0, 1\}$. Required is a protocol with the following properties:
>
> 1. Every non-faulty processor p_i irreversibly "decides" on a value $y_i \in \{0, 1\}$.
> 2. The non-faulty processors all decide on the same value.
> 3. The non-faulty processors all decide simultaneously, i.e., in the same round of computation.
> 4. If all initial bits x_i are identical, then all non-faulty processors decide x_i.

A related problem, in which condition 4 is modified to require that the non-faulty processors decide x_i only in case all processors start with x_i and no failures occur, is called *Weak Simultaneous Byzantine Agreement* (WSBA). Throughout the paper we will use t to denote an upper bound on the number of faulty processors. We call a distributed system whose processors are unreliable a *Byzantine environment*.

The Byzantine agreement problem embodies some of the fundamental issues involved in the design of effective protocols for unreliable systems, and has been studied extensively in the literature (see [F] for a survey). Interestingly, although many researchers have obtained a good intuition for the Byzantine agreement problem, many aspects of this problem still seem to be mysterious in many ways, and the general rules underlying some of the phenomena related to it are still unclear.

A number of recent papers have looked at the role of knowledge in distributed computing (cf. [CM], [HM], [PR]). They suggest that knowledge is an important conceptual abstraction in distributed systems, and that the design and analysis of distributed protocols may benefit from explicitly reasoning about the states of knowledge that the system goes through during an execution of the protocol. In [HM], special attention is given to states of knowledge of *groups* of processors, with the states of *common knowledge* and *implicit knowledge* singled out as states of knowledge that are of particular interest. As we will see, in order to be able to reach SBA on a decision value v, the non-faulty processors

must attain common knowledge that conditions that allow deciding v hold. In fact, the problem of attaining common knowledge of a given fact in a Byzantine environment turns out to be a direct generalization of the SBA problem.

We wish to investigate the states of knowledge that can be attained by the group of non-faulty processors in a Byzantine environment. In particular, we are interested in determining what facts become common knowledge at the various stages of the execution of a particular protocol. In this paper we restrict our attention to systems in which communication is synchronous and reliable, and the only type of processor faults possible are *crash failures*: a faulty processor might crash at some point, after which it sends no messages at all. Despite the fact that crash failures are relatively benign, and dealing with arbitrary possibly malicious failures is often more complicated, work on the Byzantine agreement problem has shown that many of the difficulties of working in a Byzantine environment are already exhibited in this model. By analyzing the states of knowledge that processors can attain as a function of the pattern of messages in a given protocol, we can characterize the types of coordinated simultaneous actions that can be performed at various points in the execution of the protocol. The results of this analysis directly apply to the design of protocols for SBA, WSBA, and other problems.

The main contribution of this paper is to illustrate how a knowledge-based analysis of protocols in a Byzantine environment can provide insight into the fundamental properties of such systems. This insight can be used to help us design improved t-resilient protocols for Byzantine agreement and related problems. We perform a careful analysis of the upper and lower bound proofs on the number of rounds necessary to reach common knowledge of facts in a Byzantine system. Our lower bound proofs generalize and simplify the proof of the $t+1$ round worst-case lower bound for SBA (cf. [DLM], [DS], [CD], [FL], [H], [LF]), and characterize for the first time exactly which patterns of failures require the protocol to run for $t+1$ rounds. We similarly characterize the failure patterns that allow attaining SBA in k rounds of communication, for all $k \leq t+1$, and construct a simple protocol for SBA that always halts at the earliest possible round, given the pattern in which processors fail during a given run of the protocol. In many cases, this turns out to be much earlier than in any protocol previously known.

The analysis also provides some insight into how assumptions about the reliability of the system affect the states of knowledge attainable in the system. We briefly consider some other reliability assumptions and apply our analysis to them.

Section 2 contains the basic definitions and some of the fundamental properties of our model of a distributed system and of knowledge in a distributed system. Section 3 investigates the states of knowledge attainable in a particular fairly general protocol. Section 4 contains an analysis of the lower bounds corresponding to the analysis of Section 3, simplifying and generalizing the well-known $t+1$ round worst-case lower bound for reaching SBA. Section 5 discusses some applications of our analysis to problems related to SBA, and Section 6 includes some concluding remarks.

2. Definitions and preliminary results

In this section we present a number of basic definitions that will be used in the rest of the paper, and discuss some of their implications. Our treatment will generally follow along the lines of [HM], simplified and modified for our purposes.

We consider a synchronous distributed system consisting of a finite collection of $n \geq 2$ processors (automata) $\{p_1, p_2, \ldots, p_n\}$, each pair of which are connected by a two-way communication link. The processors share a discrete global clock that starts out at time 0 and advances by increments of one. Communication in the system proceeds in a sequence of *rounds*, with round k taking place between time $k-1$ and time k. In each round, every processor first sends the messages it needs to send to other processors, and then it receives the messages that were sent to it by other processors in the same round. The identity of the sender and destination of each message, as well as the round in which it is sent, are assumed to be part of the message. At any given time, a processor's *message history* consists of the set of messages it has sent and received. Every processor p starts out with some *initial state* σ. A processor's *view* at any given time consists of its initial state, message history, and the time on the global clock. We think of the processors as following a *protocol*, which specifies exactly what messages each processor is required to send (and what other actions the processor should take) at each round, as a *deterministic* function of the processor's view. However, a processor might be *faulty*, in which case it might commit a stopping failure at an arbitrary round $k > 0$. If a processor *commits a stopping failure* at round k (or simply *fails* at round k), then it obeys its protocol in all rounds preceding round k, it does not send any messages in the rounds following k, and in round k it sends an arbitrary (not necessarily strict) subset of the messages it is required by its protocol to send. (Since a failed processor sends no further messages, we need not make any assumptions regarding what messages it receives in its failing round and in later rounds.) For technical reasons, we assume that once a processor fails, its view becomes a distinguished *failed* view. The set A of *active* processors at time k consists of all of the processors that did not fail in the first k rounds.

A *run* ρ of such a system is a complete history of its behavior, from time 0 until the end of time. This includes each processor's initial state, message history, and, if the processor fails, the round in which it fails. An *execution* is a pair (ρ, k), where ρ is a run and k is a natural number. We will use (ρ, k) to refer to the state of ρ after its first k rounds. Two executions (ρ, k) and (ρ', k) will be considered equal if all processors start in the same initial states and display the same behavior in the first k rounds of ρ and ρ'. The list of the processors' initial states is called the system's *initial configuration*. We denote processor p's view at (ρ, k) by $v(p, \rho, k)$. Furthermore, we will sometimes parameterize the set A of active processors by the particular execution, denoted $A(\rho, k)$.

Following [HM], we identify a distributed system with the set S of the possible runs of a particular fixed protocol $P = (P(1), \ldots, P(n))$, where $P(i)$ is the part of the protocol followed by processor p_i. This set essentially encodes all of the relevant information about the execution of the protocol in the system. In analyzing the properties of t-resilient protocols, the system we are interested in is the set of all possible runs of the protocol in which the system starts in one of a set of possible initial configurations, and no more than

t processors fail. Such a set will be called a *t-uniform* system for $\mathcal{P}$. A given protocol is a t-resilient protocol for SBA if all runs of the t-uniform system in which the set of possible initial configurations is $\{0,1\}^n$ satisfy the requirements of SBA.

We assume the existence of an underlying logical language for representing *ground* facts about the system. By ground we mean facts about the state of the system that do not explicitly mention processors' knowledge. Formally, a ground fact φ will be identified with a set of executions $\tau(\varphi) \subseteq S \times N$, where N is the set of natural numbers. Given a run $\rho \in S$ of the system and a time k, we will say that φ holds at (ρ,k), denoted $(S,\rho,k) \models \varphi$, iff $(\rho,k) \in \tau(\varphi)$. We will define various ground facts as we go along. The set of executions corresponding to these facts will be clear from the context.

Given a system S, we now formally define what facts a processor is said to "know" at any given point (ρ,k) for $\rho \in S$. (Our definition will correspond to [HM]'s "total view" interpretation of knowledge). We say that a processor p_i *knows* a fact ψ in S at (ρ,k), denoted $(S,\rho,k) \models K_i\psi$, if for all executions $(\rho',k) \in S \times \{k\}$ satisfying $v(p_i,\rho,k) = v(p_i,\rho',k)$ it is the case that $(S,\rho',k) \models \psi$. Roughly speaking, p_i knows ψ if ψ is guaranteed to hold, given p_i's view of the run. Notice that this definition guarantees that the "knowledge axiom" $K_i\varphi \supset \varphi$ is valid[1] (see [HM], [HM2] for other properties of K_i under this definition).

Having defined knowledge for individual processors, we now extend this definition to states of group knowledge. Given a group $G \subseteq \{p_1,\ldots,p_n\}$, we first define G's *view* at (ρ,k), denoted $v(G,\rho,k)$:

$$v(G,\rho,k) \;=\; \{\langle p, v(p,\rho,k)\rangle : p \in G\}.$$

Thus, roughly speaking, G's view is simply the joint view of its members. Extending our definition for individuals' knowledge, we say that *the group G has implicit knowledge* of φ at (ρ,k), denoted $(S,\rho,k) \models I_G\varphi$, if for all runs $\rho' \in S$ satisfying $v(G,\rho,k) = v(G,\rho',k)$ it is the case that $(S,\rho',k) \models \varphi$. Intuitively, G has implicit knowledge of φ if the joint view of G's members guarantees that φ holds. Notice that if processor p knows φ and processor q knows $\varphi \supset \psi$, then together they have implicit knowledge of ψ, even if neither of them knows ψ individually. We refer the reader to [HM] and [HM2] for a discussion and a formal treatment of I_G. In this paper we are mainly interested in states of knowledge of the group A of active processors. The set of active processors is said to implicitly know φ, denoted $I_A\varphi$, exactly if $I_G\varphi$ holds for the set $G = A$. Stated more formally,

$$(S,\rho,k) \models I_A\varphi \;\text{ iff }\; (S,\rho,k) \models I_G\varphi \;\text{ for } G = A(\rho,k).$$

Although $I_A\varphi$ is defined in terms of $I_G\varphi$, it is not the case that I_A and I_G have the same properties. The reason for this is that whereas G is a fixed set, membership in A may vary over time and differs from one run to another. Thus, for example, it is often the case that for $G = A(\rho,k)$ we have $(S,\rho,k) \not\models I_G(A = G)$, because there is some run $\rho' \in S$ such that $v(G,\rho,k) = v(G,\rho',k)$ and where G is a strict subset of $A(\rho',k)$. Consequently, whereas the formula $\neg I_G\varphi \supset I_G\neg I_G\varphi$ is valid, the corresponding formula $\neg I_A\varphi \supset I_A\neg I_A\varphi$ is not

[1] A formula is said to be *valid* if it is true of all executions in all systems.

valid! (Notice, however, that $I_A(G \subseteq A)$ holds whenever $G \subseteq A$.)[2] Since the form of implicit knowledge that concerns us most is I_A, we will call it simply *implicit knowledge*, and denote it by I.

We now show that, roughly speaking, in t-uniform systems once a fact about the past is not implicitly known it is lost forever; it will not become implicit knowledge at a later time. We say that a fact ψ is *about the first k rounds* if for all runs $\rho \in S$ it is the case that $(S, \rho, k) \models \psi$ iff $(S, \rho, \ell) \models \psi$ for all $\ell \geq k$. In particular, facts about the first 0 rounds are facts about the initial configuration. We now have:

Theorem 1: Let S be a t-uniform system, let ψ be a fact about the first k rounds, and let $\ell > k$. If $(S, \rho, k) \not\models I\psi$ then $(S, \rho, \ell) \not\models I\psi$.

Proof: Let $\ell > k$, and let ρ and ψ be such that ψ is about the first k rounds and $(S, \rho, k) \not\models I\psi$. Let $G = A(\rho, k)$. It follows that there exists a run $\rho' \in S$ such that $v(G, \rho, k) = v(G, \rho', k)$, and $(S, \rho', k) \not\models \psi$. Let ρ'' be a run with the following properties: (i) $(\rho'', k) = (\rho', k)$; (ii) All processors in $A(\rho', k) - G$ fail in round $k+1$ of ρ'' before sending any messages; and (iii) From round $k+1$ on all processors in G behave in ρ'' exactly as they do in ρ. Notice that $\rho'' \in S$ since all of the processors follow the same protocol in ρ'' and in ρ, and no more processors fail in ρ'' than do in ρ. By construction of ρ'' we have that $A(\rho'', \ell) = A(\rho, \ell)$ and that the active processors have identical views in (ρ'', ℓ) and in (ρ, ℓ). It follows that $(S, \rho'', \ell) \models I\psi$ iff $(S, \rho, \ell) \models I\psi$. Since ψ is a fact about the first k rounds and $(\rho'', k) = (\rho', k)$, we have that $(S, \rho'', \ell) \not\models \psi$ because $(S, \rho', k) \not\models \psi$. Thus, in particular, $(S, \rho'', \ell) \not\models I\psi$ and it follows that $(S, \rho, \ell) \not\models I\psi$ and we are done. $\bowtie$

Fagin and Vardi perform an interesting analysis of implicit knowledge in reliable systems (cf. [FV]). Among other things, they prove that the set of facts that are implicit knowledge about the initial configuration does not change with time. I.e., in reliable systems the implication in the statement of the Theorem 1 becomes an equivalence. However, in t-uniform Byzantine systems it is clearly the case that implicit knowledge can be "lost". For example, if processor p_i may start in initial states σ and σ', and in a particular run of the system p_i starts in state σ and fails in the first round before sending any messages, then whereas $I(\text{"}p_i \text{ started in state } \sigma\text{"})$ holds at time 0, it does not hold at any later time.

We now introduce the two other states of group knowledge that are central to our analysis. Given a group of processors G, $E_G\varphi$ (read "everyone in G knows φ") and $C_G\varphi$ ("φ is common knowledge in G") are defined as follows (cf. [HM]):

$$E_G^1\varphi = E_G\varphi = \bigwedge_{p_i \in G} K_i\varphi,$$

$$E_G^{m+1}\varphi = E_G(E_G^m\varphi), \quad m \geq 1, \qquad \text{and}$$

$$C_G\varphi = \varphi \wedge E_G\varphi \wedge E_G^2\varphi \wedge \cdots \wedge E_G^m\varphi \wedge \cdots.$$

[2] Whereas I_G satisfies the axioms of the logical system S5, it is easy to show that I_A satisfies the axioms of S4 (cf. [HM2]).

The states E_A and C_A, in which we will be most interested, are defined in the same way as E_G and C_G. Because membership in A is not explicitly given, it is sometimes useful to think of $E_A\varphi$ in the following equivalent form:

$$E_A\varphi = \bigwedge_{1\le i\le n} (p_i \in A \supset K_i\varphi),$$

It is interesting to note that in contrast to the case of implicit knowledge, the basic properties of E_A and C_A are the same as those of E_G and C_G, stated in [HM]. In particular, C_A satisfies the axioms of S5 (cf. [HM2]). Thus, in particular, C_A satisfies the "consequence closure" axiom:

CONSEQUENCE CLOSURE: $(C_A\varphi \wedge C_A(\varphi \supset \psi)) \supset C_A\psi$.

A fact that is crucial in our proofs is that C_A satisfies the "induction" axiom:

INDUCTION AXIOM: $C_A(\varphi \supset E_A\varphi) \supset (\varphi \supset C_A\varphi)$.

In the remainder of this paper, we will use I, E, and C as shorthand for I_A, E_A, and C_A.

Two executions (ρ, k) and (ρ', k) are said to be *directly similar*, denoted $(\rho, k) \approx (\rho', k)$, if for some processor p active in ρ at time k it is the case that $v(p, \rho, k) = v(p, \rho', k)$. Thus, two executions are directly similar if some active processor cannot distinguish between them. As an immediate consequence of our the definitions, we have:

$$(S, \rho, k) \models E\varphi \text{ iff } (S, \rho', k) \models \varphi \text{ for all } \rho' \in S \text{ such that } (\rho, k) \approx (\rho', k)$$

Notice that the $\approx$ relation is reflexive and symmetric, but not transitive. We say that (ρ, k) and (ρ', k) are *similar*, denoted $(\rho, k) \sim (\rho', k)$, if for some finite m there are runs $\rho_1, \rho_2, \ldots, \rho_m \in S$ such that

$$(\rho, k) \approx (\rho_1, k) \approx (\rho_2, k) \approx \cdots \approx (\rho_m, k) \approx (\rho', k).$$

The similarity relation $\sim$ is simply the transitive closure of the $\approx$ relation, and thus is an equivalence relation.

We can now show:

Theorem 2:

a) $(S, \rho, k) \models C\varphi$ iff $(S, \rho', k) \models \varphi$ for all $\rho' \in S$ such that $(\rho, k) \sim (\rho', k)$.

b) If $(S, \rho, k) \models \varphi$ for all $\rho \in S$, then $(S, \rho, k) \models C\varphi$ for all $\rho \in S$.

Proof: (a) follows by a straightforward induction on m showing that $(S, \rho, k) \models E^m\varphi$ iff $(S, \rho', k) \models \varphi$ for all ρ' such that there exist $\rho_1, \ldots, \rho_{m-1}$ with $(\rho, k) \approx (\rho_1, k) \approx \cdots \approx (\rho_{m-1}, k) \approx (\rho', k)$. Part (b) follows directly from (a). ⋈

Theorem 2 is very useful in relating common knowledge and actions that are guaranteed to be performed simultaneously. For example, we can use Theorem 2(b) and the "induction axiom" in order to relate the ability or inability to attain common knowledge of certain facts with the possibility or impossibility of reaching simultaneous Byzantine agreement. We model a processor's "deciding v" by the processor sending the message "the decision value is v" to itself, and have:

Corollary 3: Let S be a system in which the processors follow a protocol for SBA. If the active processors decide on a value v at (ρ, k), then

a) $(S, \rho, k) \models C($ "All processors are deciding v"$)$, and

b) $(S, \rho, k) \models C($"At least one processor had v as its initial value"$)$.

Proof: Let φ be the fact "all processors are deciding v". Given that the protocol guarantees that SBA is attained in S, it is the case that whenever some processor decides v all active processors do, and thus the formula $\varphi \supset E\varphi$ is valid in S (i.e., for all $\rho \in S$ and $k \geq 0$ we have $(S, \rho, k) \models \varphi \supset E\varphi$). Thus, by Theorem 2(b) it follows that $C(\varphi \supset E\varphi)$ is also valid. The "induction axiom" states that $C(\varphi \supset E\varphi) \supset (\varphi \supset C\varphi)$. Combining these two facts we have that $\varphi \supset C\varphi$ is valid, and thus if $(S, \rho, k) \models \varphi$ then $(S, \rho, k) \models C\varphi$ and we are done with part (a). For (b), let ψ be "at least one processor had v as its initial value", and notice SBA guarantees that $\varphi \supset \psi$ is valid in S. Thus, by Theorem 2(b), so is $C(\varphi \supset \psi)$. The "consequence closure" axiom states that $(C\varphi \wedge C(\varphi \supset \psi)) \supset C\psi$ is valid, and we conclude that $C\varphi \supset C\psi$ is valid. By part (a) we have that $(S, \rho, k) \models \varphi$ implies that $(S, \rho, k) \models C(\varphi)$, from which we can now conclude that $(S, \rho, k) \models C\psi$ and we are done. ⋈

3. Analysis of a simple protocol

In this section we take a close look at t-uniform systems $S_{\hat{P}}$ in which all processors follow a simple and fairly general protocol $\hat{P}$: *For $k \geq 0$, in round $k+1$ each processor sends its view at time k (i.e., after k rounds) to all other processors.*

We are interested in the states of knowledge about the initial configuration that the set of active processors attains at different stages of the execution of this protocol. Intuitively, the protocol $\hat{P}$ should provide the processors with "as much knowledge as possible" about the initial configuration, and facilitate the ability of the system to perform actions that depend on the initial configuration.

A fact φ is called *stable* if once it becomes true it remains true forever (cf. [HM]). For example, facts about the first k rounds, and in particular facts about the system's initial configuration, are stable. Since a processor's knowledge is based on a processor's view, and an active processor's view grows monotonically with time, it is the case that if φ is stable then so are $E\varphi$ and $C\varphi$ (although, as we have seen, this is not true for $I\varphi$).

A round in which no processor fails is called a *clean* round. Similarly, a round that is not clean is called *dirty*. If, for some k, round k of a run in which the processors all follow $\hat{P}$ is clean, then every active processor's view at the end of round k includes the view of the active processors at time $k-1$. In particular it follows that any stable fact that is implicit knowledge at time $k-1$ is known to everyone at time k. Consequently, at time k all processors know exactly the same facts about the initial configuration. Furthermore, Theorem 1 together with the fact that $E\varphi$ is stable when φ is, imply that at any point after a clean round, all of the processors have identical knowledge about the initial configuration. Therefore, once it is common knowledge that there was a clean round, it is common knowledge that the processors have an identical view of the initial configuration. Recall that any property that holds at all points (ρ, k) is common knowledge at all points (ρ, k).

In particular, it is common knowledge no more than t processors can fail in any run of the system, and that all processors are following the protocol $\hat{P}$. We can now show:

Theorem 4: Let φ be a fact about the initial configuration.

a) $(S_\rho, \rho, t+1) \models I\varphi$ iff $(S_\rho, \rho, t+1) \models C\varphi$.

b) $(S_\rho, \rho, n-1) \models I\varphi$ iff $(S_\rho, \rho, n-1) \models C\varphi$.

Proof: Notice that the "if" direction in both cases is immediate, since $C\psi \supset I\psi$ is valid for all facts ψ. We now show the other direction. Let φ be a fact about the initial configuration. Since at most t processors fail in any run of S_ρ, it follows by the pigeonhole principle that at least one of the first $t+1$ rounds of every run is clean. By Theorem 1 and the discussion above we have that at any point following a clean round it is the case that $I\varphi$ holds iff $E\varphi$ does. In particular, this means that in all runs of S_ρ it is the case that after $t+1$ rounds $I\varphi$ holds iff $E\varphi$ does. Notice also that $E\varphi \equiv E(I\varphi)$ is valid (since $K_i\varphi \supset I\varphi$ is). Now by Theorem 2(b) and the "induction axiom" we are done. For part (b), notice that in all runs of S_ρ one of the following two possibilities holds: either there is a clean round by time $n-1$, or there is at most one active processor at time $n-1$. In the first case we can argue as in (a) that $I\varphi$ holds at time $n-1$ iff $E(I\varphi)$ does. However, this is also true in the second case, since when there is at most one active processor p_i it is the case that $K_i\psi \equiv I\psi \equiv E\psi$. And since $K_i\psi \supset K_i(I\psi)$ is valid, for all facts ψ we have that $I\psi \equiv E(I\psi)$. Thus, again by Theorem 2(b) and the "induction axiom" we are done. ⋈

As a consequence of Theorem 4 and the discussion preceding it we have that any action that depends on the system's initial configuration can be carried out simultaneously in a consistent way by the set of active processors at any time $k \geq \min\{t+1, n-1\}$. This is consistent with the fact that there are simple t-resilient protocols for SBA that attain SBA in $t+1$ rounds. Interestingly, none of the known protocols for SBA attain SBA in less than $t+1$ rounds in *any* run. It is therefore natural to ask whether a protocol for SBA can ever attain SBA in less than $t+1$ rounds. Clearly, once it is common knowledge that a clean round has occurred, SBA can be attained. And as we shall see, there are cases in which the existence of a clean round becomes common knowledge before time $t+1$. *When* the existence of a clean round becomes common knowledge depends crucially on the pattern of failures, and on the time in which failures become implicitly known to the group of active processors. For example, if a processor p detects t failures in the first round of a run of $\hat{P}$, then the second round of the run will be clean, and at the end of the second round all active processors will know that p detected t failures in round 1. It follows from the induction axiom and Theorem 2(b) that at the end of round 2 it will be *common knowledge* that all processors have an identical view of the initial configuration (check!). Clearly, the processors can then perform any action that depends on the initial configuration (e.g., SBA) in a consistent way. In the remainder of this section we show a class of runs of S_ρ in which the processors attain common knowledge of an identical view of the initial configuration at time k, for every k between 2 and $t+1$. In the next section, we will prove that this is in fact a precise classification of the runs according to the time in which common knowledge of an identical view of the initial configuration is attained.

Intuitively, if there are more than k failures by the end of round k, then from the point of view of the ability to delay the first clean round, failures have been "wasted". In particular, if for some k it is the case that there are $k+j$ failures by the end of round k, then there must be a clean round before time $t+1-j$ (in fact, between round $k+1$ and round $t+1-j$). This motivates the following definitions: We denote the number of processors that fail by the end of round k in ρ by $N(\rho,k)$. We define the *difference at* (ρ,k), denoted $d(\rho,k)$, by

$$d(\rho,k) \stackrel{\text{def}}{=} N(\rho,k) - k.$$

We also define the *maximal difference in* (ρ,ℓ), denoted $D(\rho,\ell)$, by

$$D(\rho,\ell) \stackrel{\text{def}}{=} \max_{k\leq\ell} d(\rho,k).$$

Observe that $d(\rho,0)=0$ for all runs ρ, since $N(\rho,0)=0$. Furthermore, in a t-uniform system it is always the case that $d(\rho,k)\leq t-k$, since $N(\rho,k)\leq t$. Let D be a variable whose value at a point (ρ,k) is $D(\rho,k)$. Similarly, let $d(k)$ be a variable whose value at any point in ρ is $d(\rho,k)$. An important observation is that if at time $t+1-j$ it is common knowledge that $D\geq j$, then it is common knowledge that a clean round has occurred, and that all processors have an identical view of the initial configuration. As we will see, the protocol $\hat{P}$ has the property that if it ever becomes implicit knowledge that $D\geq j$ then at time $t+1-j$ it is common knowledge that $D\geq j$. This leads us to the following definition: Given a system S, the *wastefulness* of (ρ,ℓ) with respect to S, denoted $\mathcal{W}(S,\rho,\ell)$, is defined by:

$$\mathcal{W}(S,\rho,\ell) \stackrel{\text{def}}{=} \max\{j : (S,\rho,\ell) \models I(D\geq j)\}.$$

In words, the wastefulness of (ρ,ℓ) is the maximal value that the difference $d(\rho,\cdot)$ is implicitly known to have assumed by time ℓ. We now formally prove the claims informally stated above. We start with a somewhat technical lemma discussing the properties of wastefulness in the case of $S_{\mathcal{P}}$:

Lemma 5: Let $\rho\in S_{\mathcal{P}}$.

a) If $\mathcal{W}(S_{\mathcal{P}},\rho,\ell)=j$ then there is a particular $k\leq\ell$ such that $(S_{\mathcal{P}},\rho,\ell)\models I(d(k)\geq j)$.

b) If $I(d(k)\geq j)$ holds at time k then at time $k+1$ either $E(d(k)\geq j)$ holds, or $I(d(k+1)\geq j)$ does.

c) $\mathcal{W}(S_{\mathcal{P}},\rho,k+1)\geq\mathcal{W}(S_{\mathcal{P}},\rho,k)$ for all $k\geq 0$.

Proof: For part (a), let $\rho\in S_{\mathcal{P}}$ satisfy $(S_{\mathcal{P}},\rho,\ell)\models I(D\geq j)$, and assume that for no k is it the case that $(S_{\mathcal{P}},\rho,\ell)\models I(d(k)\geq j))$. Let ρ' be a run of $\hat{P}$ such that $(\rho',0)=(\rho,0)$, and in which the only messages not to be delivered are those that are implicitly known at (ρ,ℓ) not to have been delivered. It is easy to check that $\rho'\in S_{\mathcal{P}}$, since no more than t processors fail in it, and processor failures are crash failures. Because it is not implicit knowledge at (ρ,ℓ) that $d(k)\geq j$ for any k, it follows that $D(\rho',\ell)<j$. If we show that the group $G=A(\rho,\ell)$ has exactly the same view in (ρ,ℓ) and in (ρ',ℓ) we will be done, since this will contradict the assumption that $(S_{\mathcal{P}},\rho,\ell)\models I(D\geq j)$. We now prove that $A(\rho,\ell)$ has the same view in (ρ,ℓ) and in (ρ',ℓ). Define $G(\ell)=A(\rho,\ell)$. For $k<\ell$, assume

inductively that $G(k+1)$ is defined, and for all processors $p_i \in G(k+1)$ let $g(p_i,k)$ be the set of processors from which p_i receives a message in round $k+1$ of ρ. Define

$$G(k) \stackrel{\text{def}}{=} \bigcup_{p_i \in G(k+1)} g(p_i,k).$$

Let $G'(\ell) = G(\ell)$, and for $k < \ell$ define $g'(p_i,k)$ and $G'(k)$ from $G'(k+1)$ in an analogous fashion (substituting G, g, and ρ by G', g', and ρ'). We now show by induction on $\ell - k$ that if $k < \ell$ then for all $p_i \in G(k+1)$ we have that $g(p_i,k) = g'(p_i,k)$ and that $G(k) = G'(k)$. Let $k < \ell$ and assume inductively that $G(k+1) = G'(k+1)$. (Notice that we have defined $G(\ell) = G'(\ell)$.) Let $p_i \in G(k+1)$. The protocol $\hat{P}$ guarantees that the precise identity of $g(p_i,k)$ for $p_i \in G(k+1)$ is implicitly known at (ρ,ℓ). It follows that processor p_j sends a message to p_i in round $k+1$ of ρ iff p_j sends p_i a round $k+1$ message in ρ'. It thus follows that $g(p_i,k) = g'(p_i,k)$. Since this is true for all $p_i \in G(k+1)$, we have that $G(k) = G'(k)$, and the claim is proven. Notice that $G(k) \supseteq G(k+1)$. We now show by induction on k that for all $p_i \in G(k)$ it is the case that $v(p_i,\rho,k) = v(p_i,\rho',k)$. The case $k = 0$ follows from the fact that $(\rho,0) = (\rho',0)$ and $G(0) = G'(0)$. Assume inductively the claim holds for $k-1$, and we prove it for k. Observe that $v(p_i,\rho,k)$ for $p_i \in G(k)$ is determined by $v(p_i,\rho,k-1)$ and by $v(g(p_i,k-1),\rho,k-1)$. Since by the inductive hypothesis we have that $g(p_i,k-1) = g'(p_i,k-1)$, and that $v(g(p_i,k-1),\rho,k-1) = v(g'(p_i,\rho',k-1)$, and that $v(p_i,\rho,k-1) = v(p_i,\rho',k-1)$, it follows that $v(p_i,\rho,k) = v(p_i,\rho',k)$. It now follows that $v(G(\ell),\rho,\ell) = v(G(\ell),\rho',\ell)$, and we are done with part (a).

For part (b), assume that $(S_\rho,\rho,k) \models I(d(k) \geq j)$. If $d(k) \geq j$ is not known to everyone at $(\rho,k+1)$ then there must be (at least one) processor, say q, that fails in round $k+1$ by not sending a message to at least one processor, say p, that is active at time $k+1$. Thus, in particular, p knows at time $k+1$ that q has failed. Now, $\hat{P}$ ensures that all processors that fail by (ρ,k) are known by everyone at $(\rho,k+1)$ to have failed. It follows that if $d(k) \geq j$ is not known to everyone at time $k+1$ then $d(k+1) \geq j$ is implicit knowledge at that time. For (c), assume that $\mathcal{W}(\rho,k) = j$. Then by part (a) there is some $k' \leq k$ such that $(S_\rho,\rho,k) \models I(d(k') \geq j)$. Without loss of generality let k' be the largest such number. If $k' < k$ then by (b) we have that at time $k'+1 \leq k$ everyone knows that $d(k') \geq j$. But $E(d(k') \geq j)$ is a stable fact because $d(k') \geq j$ is, and in this case $\mathcal{W}(\rho,k+1) \geq j$, and the claim of (c) holds. If $k' = k$ then part (b) implies that at time $k+1$ either everyone will know that $d(k) \geq j$ or it will be implicit knowledge that $d(k+1) \geq j$. In both cases we will have $\mathcal{W}(\rho,k+1) \geq j$, and we are done. ⋈

Lemma 5(c) suggests that we define the *wastefulness* of a run ρ, denoted $\mathcal{W}(S,\rho)$, to be the maximal value that $\mathcal{W}(S,\rho,k)$ assumes. We now have:

Theorem 6:

a) $\mathcal{W}(S_\rho,\rho) = j$ iff $(S_\rho,\rho,t+1-j) \models E(\mathcal{W}(S_\rho,$ "the current run") $= j)$.

b) Let φ be a fact about the initial configuration. If $\mathcal{W}(S_\rho,\rho) = j$ then $(S_\rho,\rho,t+1-j) \models I\varphi$ iff $(S_\rho,\rho,t+1-j) \models C\varphi$.

Sketch of Proof: For (a), Notice that if $\mathcal{W}(S_\rho, \rho) = j$ for some $k \leq t+1-j$ it is the case that $(S_\rho, \rho, k) \models I(D \geq j)$, and at least one of the rounds $k+1, \ldots, t-j$ is clean. Lemma 5(a) and (b) imply that $I(D \geq j)$ is a stable fact in S_ρ. The claim of part(a) now follows. For (b), use part (a) to show that at $t+1-j$ the existence of a clean round is common knowledge, and follow the proof of Theorem 4. ⋈

Thus, certain patterns of failures help the processors to reach common knowledge of an identical view of the initial configuration early. As a consequence of Theorem 6 we have:

Corollary 7: There is a t-resilient protocol for SBA that reaches SBA in $t+1-\mathcal{W}(S_\rho, \rho)$ rounds in all runs ρ of the protocol in which at most t processors fail.

Proof: The protocol (identical for all processors p_i) is:

for $\ell \geq 0$ *perform the following at time* ℓ*:*
  if $K_i(D \geq t+1-\ell)$
    then *halt (and send no messages in the following rounds)*;
      decide 0 if K_i("some initial value x_j was 0");
      decide 1 otherwise.
    else *send* p_i*'s current view to all processors in round* $\ell+1$.

By Theorem 6(a) all correct processors halt after $t+1-\mathcal{W}(S_\rho, \rho)$ rounds. By Theorem 6(b) the active processors have common knowledge of the fact that they have an identical view of the initial configuration. Thus, their decisions are identical. The decision function clearly satisfies the requirements of SBA. ⋈

Notice that in runs in which many failures become visible early it is the case that SBA is attained by this protocol significantly earlier than time $t+1$. We are aware of no other protocol for SBA that stops before time $t+1$ in some cases. In the next section we will show that the protocol of Corollary 7 is optimal in the sense that for any given pattern of failures, it attains SBA no later than any other protocol for SBA does.

The number of bits of information required to describe a processor's view at round k is exponential in k. Thus, messages in the above protocol might be too long to be practical. By modifying the protocol slightly so that messages specify only the sender's view of the initial configuration and of the failure pattern, we get a protocol for SBA with the same properties in which the length of each message is $O(n + t \log n)$.

4. Lower bounds

We are about to show that the only non-trivial facts that can become common knowledge in a run ρ of a t-uniform system S before time $t+1-\mathcal{W}(S, \rho)$ are facts about the wastefulness of the run. We do this by showing that all executions (ρ, ℓ) with $\mathcal{W}(S, \rho, \ell) \leq t-\ell$ are similar. However, we first need a lemma that, roughly speaking, says that if $D(\rho, \ell) \leq t-\ell$ then (ρ, ℓ) is similar to an execution that looks just like (ρ, ℓ) (in terms of the initial configuration and the pattern of failures), except that the last processor to fail in (ρ, ℓ) never fails. More formally:

Lemma 8: Let $t \leq n-2$, and let S be a t-uniform system. Let $k \leq \ell$, let $(\rho, \ell) \in S \times \{\ell\}$ be an execution such that $D(\rho, \ell) \leq t - \ell$ and no processor fails in (ρ, ℓ) after round k. If p fails in round k of (ρ, ℓ), then there exists a run $\hat{\rho} \in S$ such that $(\rho, \ell) \sim (\hat{\rho}, \ell)$, where $(\rho, k-1) = (\hat{\rho}, k-1)$, the kth-round behavior of all processors $p' \neq p$ is identical in ρ and in $\hat{\rho}$, processor p does not fail in $(\hat{\rho}, \ell)$, and no processor fails in $(\hat{\rho}, \ell)$ after round k.

Proof: We will prove the claim by induction on $j = \ell - k$.

Case $j = 0$ (i.e., $k = \ell$): Let $Q = \{q_1, \ldots, q_s\}$ be the set of processors active at (ρ, ℓ) to whom p fails to send a message in round k of ρ. If $s = 0$ then no processor active at (ρ, ℓ) can distinguish (ρ, ℓ) from an execution $(\hat{\rho}, \ell)$ that differs from (ρ, ℓ) only in that p does not fail in $(\hat{\rho}, \ell)$. Assume that $s > 0$. Since $t \leq n-2$, there must be some processor $p_i \in A(\rho, \ell) - \{q_s\}$. Clearly, p_i's view at (ρ, ℓ) is independent of whether or not p sent a message to q_s in round ℓ. Thus, $(\rho, \ell) \sim (\rho', \ell)$, where (ρ', ℓ) differs from (ρ, ℓ) only in that p does send a message to q_s in round ℓ of (ρ', ℓ). Now, since q_s is active at (ρ', ℓ), and p does send q_s a message in round ℓ of (ρ', ℓ), processor q_s's view at (ρ', ℓ) is independent of whether p fails in (ρ', ℓ), and thus $(\rho', \ell) \sim (\hat{\rho}, \ell)$, where $(\hat{\rho}, \ell)$ has the desired properties. By transitivity of $\sim$ we also have that $(\rho, \ell) \sim (\hat{\rho}, \ell)$.

Case $j > 0$ (i.e., $k < \ell$): Assume inductively that the claim holds for $j - 1$. Again, let $Q = \{q_1, \ldots, q_s\}$ be the set of processors active at (ρ, ℓ) to whom p fails to send a message in round k of (ρ, ℓ). We prove our claim by induction on s. If $s = 0$ then no processor active in (ρ, ℓ) can distinguish whether p failed in round k or in round $k+1$. Thus, $(\rho, \ell) \sim (\rho', \ell)$, where (ρ', ℓ) differs from (ρ, ℓ) only in that rather than failing in round k, processor p fails in round $k+1$ of (ρ', ℓ) before sending any messages. Since $\ell - (k+1) = j - 1$, we have by the inductive hypothesis that $(\rho', \ell) \sim (\hat{\rho}, \ell)$, where $(\hat{\rho}, \ell)$ has the desired properties. By transitivity of $\sim$ we have that $(\rho, \ell) \sim (\hat{\rho}, \ell)$. Now assume that $s > 0$ and that the claim is true for $s - 1$. Let (ρ_s, ℓ) be an execution such that $(\rho_s, k) = (\rho, k)$, processor q_s fails in round $k+1$ of ρ_s before sending any messages, and no other processor fails in ρ_s after round k. Clearly $D(\rho_s, \ell) \leq t - \ell$, since $d(\rho_s, k') = d(\rho, k') \leq t - \ell$ for all $k' \leq k$, and $d(\rho_s, k+1) = N(\rho_s, k+1) - (k+1) = N(\rho, k) + 1 - (k+1) = d(\rho, k) \leq t - \ell$. Notice also that no processor fails in (ρ_s, ℓ) after round $k+1$. Thus, by the inductive assumption on $j - 1$, we have that $(\rho_s, \ell) \sim (\rho, \ell)$. Let $p_i \in A(\rho_s, \ell)$. Clearly p_i's view at (ρ_s, ℓ) is independent of whether p sent a message to q_s in round k of (ρ_s, ℓ). Thus, $(\rho_s, \ell) \sim (\rho'_s, \ell)$, where ρ'_s differs from ρ_s in that p does send a message to q_s in round k of ρ'_s. Again by the inductive hypothesis for $j - 1$ we have that $(\rho'_s, \ell) \sim (\rho', \ell)$, where $(\rho'_s, k) = (\rho', k)$ and no processor fails in (ρ', ℓ) after round k. Processor p fails to send round k messages only to $s - 1$ processors in ρ', and thus by the inductive hypothesis for $s - 1$ we have that $(\rho', \ell) \sim (\hat{\rho}, \ell)$, where $(\hat{\rho}, \ell)$ has the desired properties. By the symmetry and transitivity of $\sim$, we have that $(\rho, \ell) \sim (\hat{\rho}, \ell)$, and we are done. ⋈

Recall that a t-resilient protocol for SBA is required to attain SBA in all runs of the protocol in which the initial configuration is in $\{0,1\}^n$ and there are no more than t failures. Notice that in this set the initial states of the different processors are independent. We say that a t-uniform system is *independent* if the set of initial configurations possible in the system is of the form $\Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_n$, for fixed sets Σ_i. That is, there is no necessary dependence between the initial states of the different processors. We can now use Lemma 8 to show:

Theorem 9: Let $t \leq n-2$ and let S be an independent t-uniform system.

a) If $\ell \leq t$ then all failure-free executions $(\rho, \ell) \in S \times \{\ell\}$ are similar.

b) If $\mathcal{W}(S, \rho, \ell) \leq t - \ell$ and $\mathcal{W}(S, \rho', \ell) \leq t - \ell$, then $(\rho, \ell) \sim (\rho', \ell)$.

Proof: (a) Assume that $\ell \leq t$ and let (ρ, ℓ) and $(\hat{\rho}, \ell)$ be failure-free executions. We wish to show that $(\rho, \ell) \sim (\hat{\rho}, \ell)$. Let $Q = \{q_1, \ldots, q_s\}$ be the set of processors whose initial states in ρ and $\hat{\rho}$ differ. We prove by induction on s that $(\rho, \ell) \sim (\hat{\rho}, \ell)$. If $s = 0$ then $(\rho, \ell) = (\hat{\rho}, \ell)$ and we are done. Let $s > 0$ and assume inductively that all failure-free executions that differ from $(\hat{\rho}, \ell)$ in the initial state of no more than $s-1$ processors are similar to it. Let (ρ_s, ℓ) be an execution such that $(\rho, 0) = (\rho_s, 0)$, in which q_s fails in the first round without sending any messages, and no other processor fails. Clearly $D(\rho_s, \ell) = 0 \leq t - \ell$, and by Lemma 8 we have that $(\rho_s, \ell) \sim (\rho, \ell)$. Let $p_i \in A(\rho_s, \ell)$. Given that S is an independent t-uniform system, processor p_i's view at (ρ_s, ℓ) is independent of whether the initial state of q_s is as in ρ or as in $\hat{\rho}$. Thus, $(\rho_s, \ell) \sim (\rho'_s, \ell)$, where ρ'_s differs from ρ_s only in that the initial state of q_s in ρ'_s is as in $\hat{\rho}$. Again by Lemma 8 we have that $(\rho'_s, \ell) \sim (\rho', \ell)$, where $(\rho'_s, 0) = (\rho', 0)$, and (ρ', ℓ) is failure-free. Since (ρ', ℓ) differs from $(\hat{\rho}, \ell)$ only on the initial states of $s-1$ processors, by the inductive assumption we have that $(\rho', \ell) \sim (\hat{\rho}, \ell)$, and by the symmetry and transitivity of $\sim$ we have $(\rho, \ell) \sim (\hat{\rho}, \ell)$, and we are done with part (a).

(b) If $\mathcal{W}(S, \rho, \ell) \leq t - \ell$ then in particular it is not implicit knowledge at (ρ, ℓ) that $d(k) > t - \ell$ for some $k \leq \ell$. It follows that $(\rho, \ell) \sim (\bar{\rho}, \ell)$, for some $\bar{\rho} \in S$ satisfying $D(\bar{\rho}, \ell) \leq t - \ell$. Using Lemma 8, a straightforward induction on the number of processors that fail in $(\bar{\rho}, \ell)$ shows that $(\bar{\rho}, \ell) \sim (\hat{\rho}, \ell)$, where $(\hat{\rho}, \ell)$ is failure-free. By transitivity of $\sim$ we have that $(\rho, \ell) \sim (\hat{\rho}, \ell)$. The same argument applies to (ρ', ℓ), and the claim now follows from part (a). $\bowtie$

Observe that the assumption of independence of the initial configurations is essential to this lower bound. Lemma 8 can also be used to characterize non-independent systems. Lemma 8 and Theorem 9(a) generalize and somewhat simplify the $t+1$ round lower bound on the worst-case behavior of SBA in our model (see [DLM], [DS], [FL], [H], [CD]). As we will see in the sequel, Theorem 9(b) allows us to completely characterize the runs in which $t+1$ rounds are necessary for attaining SBA, as well as those that require k rounds, for all k. More generally, Theorem 2(a) and Theorem 9(b) provide us with a lower bound on the time by which facts can become common knowledge in t-uniform systems. Formally, we have:

Theorem 10: Let $t \leq n-2$, let S be an independent t-uniform system, and let $\rho' \in S$ satisfy $\mathcal{W}(S, \rho') \leq t - \ell$. If $(S, \rho', \ell) \not\models \varphi$, then $(S, \rho, \ell) \not\models C\varphi$ for all $\rho \in S$ satisfying $\mathcal{W}(S, \rho) \leq t - \ell$. $\bowtie$

Theorem 10 and Theorem 6(b) completely characterize when non-trivial facts about the initial configuration become common knowledge in the runs of $S_\mathcal{P}$. In a precise sense, they imply that the only fact that is common knowledge at (ρ, k), for $k \leq t - \mathcal{W}(S_\mathcal{P}, \rho)$, is that the wastefulness is less than $t + 1 - k$. Formally, we have:

Corollary 11: Let $t \leq n-2$, let S_ρ be an independent t-uniform system for $\hat{P}$, and let $\mathcal{W}(S_\rho, \rho) \leq t - \ell$. Then $(S_\rho, \rho, \ell) \models C\varphi$ iff for all $\rho' \in S_\rho$ such that $\mathcal{W}(S_\rho, \rho', \ell) \leq t - \ell$ it is the case that $(S_\rho, \rho', \ell) \models \varphi$. ⋈

Furthermore, Corollary 3 and Theorem 10 immediately imply:

Corollary 12: Let $t \leq n-2$, let P be a t-resilient protocol for SBA, and let S be a t-uniform system for P, with $\rho \in S$. Then SBA is not attained in ρ in fewer than $t+1-\mathcal{W}(S,\rho)$ rounds. ⋈

Corollary 12 proves that SBA cannot be attained in the runs of $\hat{P}$ any earlier than it is attained by the protocol of Corollary 7. However, it still seems possible that using another protocol SBA will be attainable in fewer rounds than in the protocol of Corollary 7. We now show that this protocol is optimal in a rather strong sense; given an initial configuration and the pattern in which failures occur, no protocol protocol attains SBA in fewer rounds than the protocol of Corollary 7. In order to state this claim rigorously and prove it, we need to make a few definitions.

We denote the initial configuration of the system by $\bar{\sigma}$. A *failure pattern* is a list π of faulty processors, and for each faulty processor p_i a specification of a round r_i in which it fails and a "forbidden" subset Q_i of the processors to whom it necessarily does not send messages in its failing round. Notice that given a protocol P, the initial configuration and failure pattern uniquely determine a run of the protocol. (However, a run of the protocol may be the result of more than one failure pattern in protocols that don't require all processors to send messages to all other processors in every round.) Thus, we can represent a run by a triple $\langle P, \bar{\sigma}, \pi \rangle$. We are now ready to show that the wastefulness of a run resulting from a given initial configuration and failure pattern is no greater than its wastefulness in S_ρ. Given Corollary 12, this will imply that the protocol of Corollary 7 always attains SBA at the earliest possible time, given the initial configuration and failure pattern.

Theorem 13: Let S be a t-uniform system for a protocol P, and let $\rho = \langle P, \bar{\sigma}, \pi \rangle$, and let $\hat{\rho} = \langle \hat{P}, \bar{\sigma}, \pi \rangle$. Then $\mathcal{W}(S,\rho) \leq \mathcal{W}(S_\rho, \hat{\rho})$.

Proof: We will show a more general fact from which the theorem will follow. Given an initial configuration $\bar{\sigma}'$, and a failure pattern π', let $\rho' = \langle P, \bar{\sigma}', \pi' \rangle$ and $\hat{\rho}' = \langle \hat{P}, \bar{\sigma}', \pi' \rangle$. Notice that $A(\rho, k') = A(\hat{\rho}, k')$ for all k'. We claim that for all k and all $p_i \in A(\rho, k)$ it is the case that if $v(p_i, \hat{\rho}, k) = v(p_i, \hat{\rho}', k)$ then $v(p_i, \rho, k) = v(p_i, \rho', k)$. We argue by induction on k. The case $k = 0$ is immediate. Let $k > 0$ and assume inductively that the claim holds for all processors in $A(\rho, k-1)$ at time $k-1$. Thus, if $v(p_i, \hat{\rho}, k) = v(p_i, \hat{\rho}', k)$ and p_j sends a round k message to p_i in $\hat{\rho}$, then p_j has the same view at $(\hat{\rho}, k-1)$ and $(\hat{\rho}', k-1)$, and p_j also sends p_i a round k message in $\hat{\rho}'$. In this case both π and π' determine that round k messages from p_j to p_i are delivered. By the inductive assumption p_j also has the same view in $(\rho, k-1)$ and in $(\rho', k-1)$. It follows that P requires p_j to act identically in round k of both ρ and ρ'. And if p_j is required to send p_i a round k message in ρ then it is required to send p_i the same message in round k of ρ'. Processor p_j does not send a round k message to p_i in $\hat{\rho}$ only if π determines that p_j cannot send p_i such a message. But then for similar reasons π' must also determine that p_j does not send p_i a round k message.

It follows that in this case p_j does not send p_i a round k message in ρ or in ρ'. Thus, for all processors p_j it is the case that p_i receives a round k message from p_j in ρ iff p_i receives an identical message from p_j in round k of ρ'. The inductive assumption also implies that $v(p_i, \rho, k-1) = v(p_i, \rho', k-1)$, and it now follows that $v(p_i, \rho, k) = v(p_i, \rho', k)$ and we are done with the claim. We now show how the theorem follows from this claim. Assume that $\mathcal{W}(S, \rho) = j$ and that $\mathcal{W}(S_\rho, \hat{\rho}) < j$. Then there is a time k such that $(S, \rho, k) \models I(D \geq j)$, and $(S_\rho, \hat{\rho}, k) \not\models I(D \geq j)$. Let $G = A(\hat{\rho}, k)$ (notice that $G = A(\rho, k)$ as well). It follows that there is a run $\hat{\rho}' \in S_\rho$ such that $v(G, \hat{\rho}, k) = v(G, \hat{\rho}', k)$ and $D(\hat{\rho}', k) < j$. Let $\bar{\sigma}'$ and π' be the initial configuration and failure pattern in $\hat{\rho}'$. Let ρ' be the run of $\mathcal{P}$ corresponding to $\bar{\sigma}'$ and π'. Since $v(G, \hat{\rho}, k) = v(G, \hat{\rho}', k)$, our claim implies that $v(G, \rho, k) = v(G, \rho', k)$. But since $D(\rho', k) = D(\hat{\rho}', k) < j$ and $A(\rho, k) = G$, we have that $(S, \rho, k) \not\models I(D \geq j)$, contradicting our original assumption. ⋈

Theorem 13 and Corollary 12 now imply that the protocol of Corollary 7 is indeed optimal in the strong sense we intended: given any initial configuration and failure pattern, it attains SBA as early as any t-resilient protocol for SBA can. In light of Theorem 13, we can talk about the inherent *wastefulness* $w(\pi)$ of a failure pattern π, defined to be $\mathcal{W}(S_\rho, \langle \hat{P}, \bar{\sigma}, \pi \rangle)$. That $w(\pi)$ is well defined follows from the fact that runs ρ of S_ρ have the property that $\mathcal{W}(S_\rho, \rho, k)$ depends only on the pattern of failures and is independent of the initial configuration. This can be proved by a straightforward induction on k, and is left to the reader. Lemma 8 through Corollary 12 can now be viewed as statements about the effect of the failure pattern on the similarity of executions and on what facts can become common knowledge at various times in the execution of an arbitrary t-resilient protocol. Corollaries 7 and 12 tell us that exactly $t + 1 - w(\pi)$ rounds are necessary to attain SBA in runs of any t-resilient protocol for SBA that have pattern failure π (in the rest of the paper we will use π to refer to the failure pattern of the run in question). This provides a complete characterization of the number of rounds required to reach SBA in a run, given the pattern in which failures occur.

We have seen that the only facts that become common knowledge before time $t + 1 - w(\pi)$ are facts about the wastefulness of the run. In the previous section we saw that in runs of S_ρ the processors attain common knowledge of an identical view of the initial configuration at time $t + 1 - w(\pi)$. Thus, we have a complete description of when facts about the initial configuration become common knowledge. It is interesting to ask the more general question of when arbitrary facts become common knowledge. Using Lemma 5 it is possible to show that at time $t + 1 - w(\pi)$ in a run of S_ρ it is not only common knowledge that there was a clean round, but there is a particular round that is commonly known to have appeared clean to all active processors. Let $k(\pi)$ denote the latest such round. Thus, at time $t + 1 - w(\pi)$ it is common knowledge that the processors have an identical view of the first $k(\pi) - 1$ rounds. There is some number, say f of processors that are commonly known at time $t+1-w(\pi)$ to have failed by time $k(\pi)-1$. Let $t' = t-f$. Roughly speaking, time $k(\pi)$ can now be regarded as the start of a new run, and for appropriate definitions of $d'(k)$ and $w'(\pi)$, we get that at time $k(\pi) + t' + 1 - w'(\pi)$ the system will attain common knowledge of a common view of the state of the system at time $k(\pi)$. Interestingly, it can be shown that $k(\pi) + t' + 1 - w'(\pi) = t + 2 - w(\pi)$. That is, one round after the processors attain common knowledge of (a common view of) the state of the run at time $k(\pi) - 1$,

they attain common knowledge of the state of the run at time $k(\pi)$. In fact, at the end of each round following time $t+1-w(\pi)$ the active processors attain common knowledge of a common view of at least one (sometimes more) additional past round. Again, the techniques of Sections 3 and 4 can be used to show that the pattern of failures determines when an arbitrary fact about the first k rounds may become common knowledge, and the simple protocol $\hat{P}$ of Section 3 is in a precise sense the fastest to attain common knowledge of such facts. Details are left to the reader.

5. Applications

Throughout the paper we have shown how our results regarding when common knowledge of various facts is attained in a Byzantine system affect the SBA problem. In this section we discuss some further consequences of the analysis presented in the previous sections. This is intended to illustrate the types of applications that the analysis can be used for. We start by considering some problems that are closely related to SBA.

The problem of *Weak* SBA (WSBA) mentioned in the introduction, which differs from SBA in that clause (4) is changed so that the active processors are required to decide on a value v only if all initial values were v and *no processor fails* was introduced by Lamport as a weakening of SBA. However, Theorem 9(b) immediately implies that the active processors do not have common knowledge of whether any processors failed before time $t+1-w(\pi)$, in any run of a t-resilient protocol for WSBA with failure pattern π. And since SBA can already be performed at time $t+1-w(\pi)$, we have that t-resilient protocols cannot attain WSBA any earlier than they can SBA. Theorem 9 also describes why the variant of SBA used in this paper (which was introduced by [FL]) is essentially equivalent to the original version of the *Byzantine Generals* problem of [PSL], in which only one processor initially has a value, and the processors need to decide on this value if the processor does not fail, and on a consistent value otherwise.

It has been a folk conjecture that a t-resilient protocol that guarantees to perform any non-trivial action simultaneously requires $t+1$ rounds in the worst case. We will now show that this is not the case. Let *Bivalent Agreement* be defined by clauses (1)–(3) of SBA, and replacing clause (4) by:

4'. At least one run of the protocol decides 0, and at least one run decides 1.

Thus, so long as no more than t processors fail, all processors must decide on the same value simultaneously, and both 0 and 1 must be attainable values. Theorem 2 implies that any action that is guaranteed to be performed simultaneously requires some fact to become common knowledge before the action can be performed. Theorem 6(b) implies that at the end of round 2 of S_ρ it is common knowledge whether or not the wastefulness of the run is $t-1$ (i.e., whether t processors were seen to have failed in the first round). Thus, we can easily derive a t-resilient protocol for Bivalent agreement: Each processor follows $\hat{P}$ for the first two rounds, and then decides 0 if it knows that t processors failed in the first round, and 1 otherwise. This protocol attains Bivalent agreement in two rounds, and Theorem 10 implies that there is no faster protocol for Bivalent agreement so long

as $t \leq n-2$. Furthermore, it implies that in a precise sense this is the only two-round protocol for Bivalent agreement. We leave it to the reader to check that if $t \geq n-1$ then there is a protocol for Bivalent agreement that requires only one round. Thus, Bivalent agreement is a truly easier problem than SBA. We note that [FLP] and [DDS] prove that in an asynchronous system there is no 1-resilient protocol for an even weaker variant of Bivalent agreement.

We have stressed the connection between common knowledge and simultaneous actions. Interestingly, the lower bounds on the time required for attaining common knowledge imply worst-case bounds on the behavior of t-resilient protocols that perform coordinated actions that are not required to be performed simultaneously. For example, *Eventual Byzantine Agreement* (EBA) is defined by clauses (1), (2), and (4) of SBA: the processors' decisions need not be simultaneous (cf. [DRS]). There are well-known protocols that attain EBA after two rounds in failure-free runs (for which $w(\pi) = 0$). However, using Theorems 2, 10, and 13 it is not hard to show that a t-resilient protocol for EBA must require $t+1$ rounds in some runs with $w(\pi) = 0$. More generally, these theorems show that such a protocol must require $t+1-j$ rounds in some runs with $w(\pi) = j$. This is a slight refinement of the well-known fact that EBA requires $t+1$ rounds in the worst case (cf. [DRS]). Many very relevant and interesting aspects of EBA are not covered by our analysis. We believe that an analysis of EBA should involve a study of when the states of ϵ-common knowledge and eventual common knowledge (cf. [HM]) are attained in a Byzantine environment. This is an interesting open problem.

As our investigation centered around t-resilient protocols, we now briefly discuss some other possible reliability assumptions. Recall that Theorem 4 states that all active processors are guaranteed to have an identical view of the system's initial configuration at time $t+1$ in every run of a t-uniform system for $\hat{P}$. This follows simply from the fact that at time $t+1$ it is common knowledge that one of the previous rounds was clean. Instead of t-resiliency, we could require that a protocol for SBA be guaranteed to attain SBA so long as no more than k consecutive rounds are dirty. In the system corresponding to all the runs of $\hat{P}$ in which at most k consecutive rounds are dirty, it is common knowledge at time $k+1$ that a clean round has occurred, and $\hat{P}$ can be converted in to a protocol for SBA that is guaranteed to attain SBA in no more than $k+1$ rounds. This means, for example, that if processors in a Byzantine system are known to fail at least two at a time, SBA can be achieved in $t/2+1$ rounds. Having a bound of k consecutive dirty rounds seems in many cases to be a more appropriate assumption about a system than having a bound of t on the total number of failures possible, since the latter is not a local assumption. Of course, these two assumptions are not mutually exclusive, and we may often have a small bound on the possible number of consecutive dirty rounds, when only a much larger bound holds for the total number of failures. The bound on the number of consecutive dirty rounds implies a good upper bound on SBA in the case of crash failures.

Another way we can consider varying the reliability assumptions about the system is by restricting the number of possible processor failures that can occur in a round. For example, let us consider the assumption that at most one processor can fail in any given round of the computation, and at most t processors might fail overall. We are interested in

the question of whether such assumptions allow us to attain SBA quickly. Unfortunately, the lower bound proofs of Lemma 8 and Theorem 9 work very well for this reliability model. In fact, since all of the runs of such a system are guaranteed to have wastefulness 0, even Bivalent agreement cannot be attained in any run of the system in less than $t + 1$ rounds! SBA and WSBA clearly require $t + 1$ rounds in *all* runs of the system. We now present a somewhat artificial variant of this assumption that provides us with a non-uniform reliability assumption whose behavior is interesting and somewhat counter-intuitive: We say that a protocol for SBA is *one visible failure resistant* (1-VFR) if it is guaranteed to attain SBA so long as no more than one processor failure becomes visible to the active processors in any given round. The set of possible runs of a protocol P that display such behavior will be called a *visibly restrained* system for P. It is possible to show that in the visibly restrained system for the simple protocol $\hat{P}$ of Section 3 it is common knowledge at time 2 whether round 1 is clean, and therefore WSBA can be attained in two rounds. However, SBA can be shown to require $n - 1$ rounds in runs of $\hat{P}$ in which one processor fails in every round except possibly the $(n - 1)$st round. (If one adds a bound of $t \leq n - 2$ on the total number of failures possible, $n - 1$ is replaced by $t + 1$.) Interestingly, there is a 1-VFR protocol for SBA that is guaranteed to attain SBA in three rounds (in all runs)! Thus, for the 1-VFR reliability model, our simple protocol is no longer a most general protocol. The reason for the odd behavior of 1-VFR protocols is that the patterns of failures of the runs that satisfy 1-VFR are intimately related to the structure of the protocol. Thus, the protocol can restrict the patterns of failures possible and make effective use of the 1-VFR assumption. Details and further discussion are given in the full paper.

6. Conclusions

This paper analyzes the states of knowledge attainable in the course of the execution of various protocols in the system, for the case of a particular simple model of unreliable distributed systems that is fairly popular in the literature. Motivated by the work of [HM], the analysis focused mainly on when facts that are implicitly known become common knowledge in systems in which there is an upper bound of t on the number of possible faulty processors. This problem was shown to be a direct generalization of problems such as Simultaneous Byzantine Agreement, in which it is required that consistent actions be performed simultaneously at all non-faulty sites of the system. By deriving exact bounds on the question of when facts become common knowledge, we immediately got exact bounds for SBA and many other problems. An interesting fact that came out of the analysis was that the pattern in which processors fail in a given run determines a lower bound on the time in which facts about the system's initial configuration become common knowledge, with different patterns determining different bounds. Ironically, facts become common knowledge faster in cases when many processors fail early in the run. The somewhat paradoxical argument for this is that, given an upper bound on the total number of failures possible, if many processors fail early then only few can fail later. The protocol can make use of the fact that the rest of the run is relatively free of failures. As a by-product of the analysis, we were able to derive a simple improved protocol for SBA that is optimal in *all* runs.

Our analysis shows that the essential driving force behind many of the phenomena in unreliable systems seems to be the inherent uncertainty that a particular site in such a system has about the global state of the system. We come to grips with this uncertainty by performing a knowledge-based analysis of such a system. We stress that our analysis was by and large restricted to protocols for simultaneous actions in a rather clean and simple model of unreliable systems: synchronous systems with global clocks and crash failures. We believe that performing similar analyses for nastier models of failures will prove very exciting, and will provide a much better understanding of the true structure underlying the richer failure models, and of the differences between the failure models. The ideas and techniques developed in this paper should provide a sound basis on which to build such an analysis, although it is clear that a number of additional ideas would be required.

In summary, the treatment in this paper differs from the usual approach to Byzantine agreement type problems in that we make explicit and essential use of reasoning about knowledge in order to reach conclusions about the possibility or impossibility of carrying out certain desired actions in a distributed environment. The generality and applicability of our results suggest that this is a promising approach.

Acknowledgements: We wish to thank Brian Coan, Ron Fagin, Joe Halpern, Nancy Lynch, and Moshe Vardi for stimulating discussions. The work of the second author was supported in part by an IBM Post-doctoral fellowship. Some of the work was done while he was at Stanford University, supported by DARPA contract N00039-82-C-0250, and by an IBM Research Student Associateship.

References

[CD] B. Coan and C. Dwork, Simultaneity is harder than agreement, To appear, *Proceedings of the Fifth Symposium on Reliability in Distributed Software and Database Systems*, 1986.

[CM] K. M. Chandy and J. Misra, How processes learn, *Proceedings of the Fourth ACM Symposium on the Principles of Distributed Computing*, 1985, pp. 204-214.

[DLM] R. DeMillo, N. A. Lynch, and M. Merritt, Cryptographic Protocols, *Proceedings of the Fourteenth Annual ACM Symposium on the Theory of Computing*, 1982, pp. 383-400.

[DDS] D. Dolev, C. Dwork, and L. Stockmeyer, On the minimal synchronization needed for distributed consensus, *Proceedings of the 24th Annual Symposium on Foundations of Computer Science*, 1983, pp. 369-397.

[DRS] D. Dolev, R. Reischuk, and H. R. Strong, Eventual is earlier than immediate, *Proceedings of the 23th Annual Symposium on Foundations of Computer Science*, 1982, pp. 196-203.

[DS] D. Dolev H. R. Strong, Polynomial algorithms for multiple processor agreement, *Proceedings of the Fourteenth Annual ACM Symposium on the Theory of Computing*, 1982, pp. 401-407.

[FV] R. Fagin and M. Y. Vardi, Knowledge and implicit knowledge in a distributed environment, *Proceedings of the Conference on Theoretical Aspects of Reasoning About Knowledge*, Monterey, 1986.

[F] M. J. Fischer, The consensus problem in unreliable distributed systems (A brief survey), *Yale University Technical Report YALEU/DCS/RR-273*, 1983.

[FL] M. J. Fischer and N. A. Lynch, A lower bound for the time to assure interactive consistency, *Information Processing Letters*, **14**:4, 1982, pp. 183-186.

[FLP] M. J. Fischer, N. A. Lynch, and M. Paterson, Impossibility of distributed consensus with one faulty process, *Proceedings of the second Symposium on Principles of Database Systems*, 1983.

[H] V. Hadzilacos, A lower bound for Byzantine agreement with fail-stop processors, *Harvard University Technical Report TR-21-83*.

[HM] J. Y. Halpern and Y. Moses, Knowledge and common knowledge in a distributed environment, Version of December 1985 is available as an IBM RJ. Early versions appeared in *Proceedings of the Third ACM Symposium on the Principles of Distributed Computing*, 1984, pp. 50–61; revised as IBM research report *RJ 4421*, 1984.

[HM2] J. Y. Halpern and Y. Moses, A guide to the modal logic of knowledge and belief, *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1985, pp. 480-490.

[LF] L. Lamport and M. J. Fischer, Byzantine grenerals and transaction commit protocols, *SRI Technical Report Op.62*, 1982.

[PR] R. Parikh and R. Ramanujam, Distributed processes and the logic of knowledge (preliminary report), *Proceedings of the Workshop on Logics of Programs*, 1985, pp. 256-268.

[PSL] M. Pease, R. Shostak, and L. Lamport, Reaching agreement in the presence of faults, *JACM*, **27**:2, 1980, pp. 228-234.