Knowledge and Common Knowledge in a Byzantine Environment: Crash failures

Cynthia Dwork

IBM Almaden Research Center,
San Jose, CA 95120

Yoram Moses

MIT Laboratory for Computer Science,
Cambridge, MA 02139

July 1986

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. In particular, we study what facts become *common knowledge* at various points in the execution of protocols in an unreliable system. This characterizes the simultaneous actions that can be carried out in such systems. For example, we obtain a complete characterization of the number of rounds required to reach *Simultaneous Byzantine Agreement*, given the pattern in which failures occur. From this we derive a new protocol for this problem that is optimal *in all runs*, rather than just always matching the worst-case lower bound. In some cases this protocol attains Simultaneous Byzantine Agreement in as few as 2 rounds. We also present a non-trivial simultaneous agreement problem called *bivalent agreement* for which there is a protocol that always halts in two rounds. Our analysis applies to simultaneous actions in general, and not just to Byzantine agreement. The lower bound proofs presented here generalize and simplify the previously known proofs.

Keywords: common knowledge, simultaneous actions, crash failure, optimal in all runs.

The work of the second author at MIT was supported in part by an IBM Postdoctoral fellowship, by the Office of Naval Research under contract N00014-85-K-0168, by Office of Army Research under contract DAAG29-84-K-0058, by the National Science Foundation under grant DCR-8302391, and by the Defense Advanced Research Projects agency (DARPA) under contract N00014-83-K-0125. 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.

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 values x_i are identical, then all non-faulty processors decide x_i.

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 for 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. Among other things, they show that common knowledge is intimately related to *simultaneous actions* — actions that are guaranteed to take place simultaneously at all sites of the system. As we shall see, processors running a protocol for SBA can decide on a particular value v

only once certain facts about the inital values x_i become common knowledge. 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.

This paper studies the structure and properties of t-resilient protocols that perform simultaneous actions by investigating what facts can become common knowledge at different points in the execution of a t-resilient protocol. 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. In the sequel we will use SBA as our standard example of a desirable simultaneous action.

Our analysis provides new insight into the basic issues involved in performing simultaneous actions in a Byzantine environment. For example, it shows that the pattern in which failures occur completely determines the number of rounds required to attain common knowledge of facts about the initial state of the system. Consequently, we obtain a complete characterization of the patterns of failures that require a t-resilient protocol for SBA to take k rounds, for $2 \leq k \leq t+1$. This generalizes the well-known fact that SBA requires $t-1$ rounds in the worst case (cf. [DLM],[DS],[CD],[FL],[H],[LF]). Our proof is a simplification of the well-known lower bound proof for SBA. Interestingly, our analysis immediately suggests a protocol for SBA that is *optimal in all runs*. That is, it halts as early as possible, given the pattern in which failures occur. In many cases, this turns out to be much earlier than in any protocol previously known. This is the first protocol for SBA that is optimal in all runs. In fact, it is the first protocol for SBA that *ever* halts before the end of round $t+1$. The $t+1$ round lower bound on the worst case behavior of protocols for SBA has often been misinterpreted to mean that SBA cannot ever be reached in less than $t+1$ rounds.

The analysis presented in this paper applies to a large class of simultaneous actions, not only to SBA. For example, we present the *bivalent agreement* problem, in which clause (4) of SBA is replaced by a requirement that the protocol have at least one run in which the processors decide 0, and at least one run in which they decide 1. We derive a protocol that always reaches bivalent agreement in two rounds. This contradicts a "folk conjecture" in the field that states that performing any non-trivial task simultaneously in a byzantine environment requires $t+1$ rounds in the worst case.

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 is very useful in the design of improved t-resilient protocols for Byzantine agreement and many related problems. 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 is 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* (sometimes also called a *point*) 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 $A(\rho, k)$.

We will find it useful to talk about the pattern in which failures occur in a given run. Formally, a *failure pattern* π is a set of triples of the form $\langle p, k(p), Q(p)\rangle$, where p is a processor, $k(p)$ is a round number, and $Q(p)$ is a set of processors. A run ρ *displays* (or, more precisely, *is consistent with*) *the failure pattern* π if (i) every processor that fails in ρ is the first element of some triple in π, and (ii) for every triple $\langle p, k(p), Q(p)\rangle$ in π it is the case that processor p fails in round $k(p)$ of ρ, in round $k(p)$ it sends no messages to processors in $Q(p)$, and it does send messages to all processors not in $Q(p)$ to which the protocol prescribes it to send. A protocol $\mathcal{P}$, initial configuration σ, and failure pattern π uniquely determine a run. (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.) We denote this run by $\mathcal{P}(\sigma, \pi)$.

Following [HM], we identify a distributed system with the set S of the possible runs of a particular fixed protocol $\mathcal{P} = (P(1), \ldots, P(n))$, 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. Given a system S, for $1 \leq i \leq n$ let Σ_i be the set of initial states that processor p_i assumes in the runs of S. The system S is said to be a *t-uniform system for $\mathcal{P}$* if there is a set of initial configurations $\Gamma \subseteq \Sigma_1 \times \cdots \times \Sigma_n$ such that S is the set of all runs of the protocol $\mathcal{P}$ starting in initial configurations from Γ in which at most t processors fail. t-uniform systems have the property that a processor failure is an event that is independent of the initial configuration and of the time in which other processors fail. A system is said to be *independent* if its set of initial configurations is of the form $\Sigma_1 \times \cdots \times \Sigma_n$. In an independent t-uniform system there is no necessary dependence between the initial states of the different processors. The properties of t-resilient protocols can be studied by analyzing particular t-uniform systems for them. For example, a given protocol is a t-resilient protocol for SBA if all runs of the independent 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 $(S, \rho, k) = \varphi$, 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. We close this language under the standard boolean connectives $\wedge$, $\neg$ and $\supset$, interpreted as the standard conjunction, negation and implication.

Given a system S, we now define what facts a processor is said to "know" at any given point (ρ, k) for $\rho \in S$. Roughly speaking, p_i is said to know a fact ψ if ψ is guaranteed to hold, given p_i's view of the run. More formally, given a system S, we say that two points (ρ, k) and (ρ', k') are *p_i-equivalent* relative to S, denoted $(\rho, k) \overset{i}{\leftrightarrow} (\rho', k')$, iff $\rho, \rho' \in S$ and $v(p_i, \rho, k) = v(p_i, \rho', k')$. (The only case in which $v(p_i, \rho, k) = v(p_i, \rho', k')$ is possible for

$k' \neq k$ is when $v(p_i, \rho, k) = \textit{failed}$.) 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 $(S, \rho', k') \models \psi$ for all executions $(\rho', k') \in S \times N$ satisfying $(\rho, k) \overset{i}{\leftrightarrow} (\rho', k')$. This definition of knowledge is essentially the *total view* interpretation of [HM]. We are about to review some of the properties of knowledge under this definition. Other properties will be covered in the sequel (see also [HM] and [HM2]).

A formula is said to be *valid* if it is true of all executions in all systems. Given a system S, a formula is said to be *valid in* S if it true of all executions of S. It follows that a valid fact is valid in S for all systems S. We now show that under our definition of knowledge, K_i satisfies the axioms of the modal system S5. This fact will follow in a straightforward way from the fact that knowledge is determined by the $\overset{i}{\leftrightarrow}$ relations, which in our case are equivalence relations.

Proposition 1:

a) If φ is valid in S then $K_i\varphi$ is valid in S.

b) The *consequence closure* axiom is valid:

$$\text{CONSEQUENCE CLOSURE:} \qquad (K_i\varphi \wedge K_i(\varphi \supset \psi)) \supset K_i\psi.$$

c) The *knowledge axiom* is valid:

$$\text{KNOWLEDGE AXIOM:} \qquad K_i\varphi \supset \varphi.$$

d) The *positive introspection* axiom is valid:

$$\text{POSITIVE INTROSPECTION:} \qquad K_i\varphi \supset K_iK_i\varphi.$$

e) The *negative introspection* axiom is valid:

$$\text{NEGATIVE INTROSPECTION:} \qquad \neg K_i\varphi \supset K_i\neg K_i\varphi.$$

Proof: For part (a), let (ρ, k) be an (arbitrarily chosen) execution satisfying $\rho \in S$, and let φ be a formula that is valid in S. Thus φ is true of all executions $(\rho', k') \in S \times N$, and, in particular, φ is true of all executions in $S \times N$ that are p_i-equivalent to (ρ, k). It thus follows that $K_i\varphi$ is true of (ρ, k), and since (ρ, k) was an arbitrary execution in $S \times N$, we have that $K_i\varphi$ is valid in S. For (b), let $(S, \rho, k) \models K_i\varphi \wedge K_i(\varphi \supset \psi)$. Then by the definition of $(S, \rho, k) \models K_i\varphi$ we have that both φ and $(\varphi \supset \psi)$ hold at all points (ρ', k) that are p_i-equivalent to (ρ, k). It thus follows that ψ holds at all such points (ρ', k), and again by the definition of $(S, \rho, k) \models \psi$ we are done. Part (c) follows from the fact that $(\rho, k) \overset{i}{\leftrightarrow} (\rho, k)$, i.e., p_i-equivalence is reflexive. Now, by definition we have that if $K_i\varphi$ is true of (ρ, k) then φ is true of all executions that are p_i-equivalent to (ρ, k), and in particular φ is true of (ρ, k). For part (d), let $(S, \rho, k) \models K_i\varphi$. Thus, φ is true of all executions $(\rho'', k'') \overset{i}{\leftrightarrow} (\rho, k)$. We wish to show that $(S, \rho', k') \models K_i\varphi$ for all

$(\rho', k') \stackrel{i}{\leftrightarrow} (\rho, k)$. Since $\stackrel{i}{\leftrightarrow}$ is an equivalence relation, all executions $(\rho'', k'') \in S \times N$ satisfy that $(\rho, k) \stackrel{i}{\leftrightarrow} (\rho'', k'')$ iff $(\rho', k') \stackrel{i}{\leftrightarrow} (\rho'', k'')$. It thus follows that φ is true of all executions $(\rho'', k'') \stackrel{i}{\leftrightarrow} (\rho', k')$, and we are done. The argument for part (e) is similar. If $(S, \rho, k) \models \neg\varphi$ then $(S, \rho, k) \not\models \varphi$, and therefore there must be an execution (ρ'', k'') that is p_i-equivalent to (ρ, k) of which φ is not true. Let (ρ', k') be an execution that is p_i-equivalent to (ρ, k). Because p_i-equivalence is an equivalence relation, we have that $(\rho', k') \stackrel{i}{\leftrightarrow} (\rho'', k'')$, and hence $(S, \rho', k') \models \neg K_i\varphi$. It now follows that $(S, \rho, k) \models K_i \neg K_i \varphi$ and we are done. $\asymp$

Roughly speaking, clauses (a) through (e) characterize the modal system S5. An operator satisfying clauses (a) through (d) is said to satisfy the modal system S4 (cf. [HM2]). An interesting consequence of our choice of having a failed processor's view be a distinguished *failed* view is the fact that a processor always knows whether it is active. Furthermore, the only things that a failed processor knows are the consequences of the fact that the processor has failed and of the formulas that are valid in S. Given that a failed processor is "out of the game" in our model, we will focus our attention on the knowledge of the active processors.

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 $v(G, \rho, k)$:

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

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. An identical proof to that of Proposition 1 now shows:

Proposition 2: The operator I_G satisfies the modal system S5 (clauses (a) through (e) of Proposition 1, substituting I_G for K_i). $\bowtie$

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. We say that *the set of active processors implicitly knows* φ, denoted $I\varphi$, exactly if $I_G\varphi$ holds for the set $G = A$. Stated more formally,

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

Although $I\varphi$ is defined in terms of $I_G\varphi$, it is not the case that I 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 negative introspection axiom for $I_{i,}$, i.e., $\neg I_{i,}\varphi \supset I_{i,}\neg I_{i,}\varphi$, is valid, the corresponding formula for I: $\neg I\varphi \supset I\neg I\varphi$, is not valid! (Notice, however, that $I(G \subseteq A)$ holds whenever $G \subseteq A$). For example, it may be the case that processor p_j sends processor p_i a message in round 1 stating p_j's initial state, and fails before sending any other message, and that processor p_i fails in round 1 after sending all of its round 1 messages. Processor p_j's initial state is thus not implicitly known to the set of active processors, but it is consistent with the active processors' joint view that p_i is active, in which case p_j's initial state would be implicitly known. The above discussion can be summarized by:

Proposition 3: The implicit knowledge operator I satisfies the modal system S4 (i.e., clauses (a) — (d) of Proposition 1). The negative introspection axiom is not valid for I. $\bowtie$

The following lemma describes the relationship between K_i and I:

Lemma 4: Let φ be a formula and let $p_i \in A(\rho, k)$.

a) If $(S, \rho, k) \models K_i\varphi$ then $(S, \rho, k) \models I\varphi$.

b) If $(S, \rho, k) \models K_i\varphi$ then $(S, \rho, k) \models K_iI\varphi$.

Proof: For part (a), assume that $(S, \rho, k) \models K_i\varphi$, and let (ρ', k') be an execution satisfying $v(A(\rho, k), \rho', k') = v(A(\rho, k), \rho, k)$. In particular, since $p_i \in A(\rho, k)$ we have that $v(p_i, \rho', k') = v(p_i, \rho, k)$, and thus since $K_i\varphi$ holds at (ρ, k), we have that φ holds at (ρ', k'). Since this is true for all such executions (ρ', k'), we are done by the definition of $(S, \rho, k) \models I\varphi$. For (b), let $(S, \rho, k) \models K_i\varphi$. By Proposition 1 (d) we have that $(S, \rho, k) \models K_iK_i\varphi$. The fact that $p_i \in A(\rho, k)$ implies that $v(p_i, \rho, k) \neq \mathit{failed}$. Thus, p_i is an active processor in all executions that are p_i-equivalent to (ρ, k). Let $(\rho', k') \overset{i}{\leftrightarrow} (\rho, k)$. We thus have that $p_i \in A(\rho', k')$, and that $K_i\varphi$ holds at (ρ', k'). Part (a) therefore implies that $I\varphi$ holds at (ρ', k'), and thus $K_iI\varphi$ holds at (ρ, k). $\bowtie$

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 5: 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. By construction, the number of processors that fail by time k in ρ'' is no larger than the number in ρ, and exactly the same processors fail in ρ and in ρ'' by any later time. Given that S is a t-uniform system and $\rho \in S$, no more than t processors fail in ρ. It follows that $\rho'' \in S$, since all of the processors follow the same protocol in ρ'' and in ρ, and no more than t processors fail in ρ''. By construction of ρ'' we also 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. $\asymp$

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 5 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. We define "everyone knows" and "common knowledge" along the lines of [HM]. In our case, however, these notions will be defined for the set of active processors. *Every (active) processor knows* φ, denoted $E\varphi$, is defined by

$$E\varphi \stackrel{\text{def}}{=} \bigwedge_{1 \le i \le n} (p_i \in A \supset K_i\varphi).$$

An immediate corollary of Lemma 4 which we will find useful in the sequel is:

Corollary 6: $E\varphi \supset E(I\varphi)$ is valid.

$\bowtie$

We define $E^1\varphi \stackrel{\text{def}}{=} E\varphi$, and $E^{m+1}\varphi \stackrel{\text{def}}{=} E(E^m\varphi)$ for $m \ge 1$. A fact φ is said to be *common knowledge among the active processors*, denoted $C\varphi$, if $E^m\varphi$ holds for all $m \ge 1$. More formally,

$$C\varphi = \varphi \wedge E\varphi \wedge E^2\varphi \wedge \cdots \wedge E^m\varphi \wedge \cdots.$$

Common knowledge among the active processors, which we will call simply common knowledge, will play a crucial role in the sequel. We now study some of its properties. A useful tool for thinking about $E^m\varphi$ and $C\varphi$ is the labelled undirected graph whose nodes are the executions of a system S, and whose edges are the $\stackrel{i}{\leftrightarrow}$ relations, restricted so that an edge $e \stackrel{i}{\leftrightarrow} e'$ exists only if p_i is active in e (and hence also in e'). (This graph is precisely the *Kripke structure* modelling the active processors' knowledge in the system; cf. [HM2].) The *distance* between two executions $e = (\rho, k)$ and $e' = (\rho', k)$ in this graph, denoted $\delta(e, e')$, is simply the length of the shortest path in the graph connecting e and e'. If there is no path connecting e to e', then $\delta(e, e')$ is defined to be infinity. Two executions e and e' are said to be *similar*, denoted $e \sim e'$ if $\delta(e, e')$ is finite (i.e., if e' and e are in the same connected component of the graph). Equivalently, $(\rho, k) \sim (\rho', k)$, if for some finite m there are runs $\rho_1, \rho_2, \ldots, \rho_{m-1} \in S$, and processors $p_{i_1}, p_{i_2}, \ldots, p_{i_m}$, satisfying $p_{i_j} \in A(\rho_j, k)$ for $j \le m-1$, $p_{i_m} \in A(\rho', k)$, and

$$(\rho, k) \stackrel{i_1}{\leftrightarrow} (\rho_1, k) \stackrel{i_2}{\leftrightarrow} \cdots \leftrightarrow (\rho_{m-1}, k) \stackrel{i_m}{\leftrightarrow} (\rho', k).$$

(The system S is usually clear from context, and thus we do not add a subscript S to *sim.*) It is now easy to check that $(S, \rho, k) \models E\varphi$ iff $(S, \rho', k) \models \varphi$ for all executions (ρ', k) of distance ≤ 1 from (ρ, k). Notice that similarity is an equivalence relation. We can now show:

Proposition 7:

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 φ is valid in S then $C\varphi$ is valid in S.

c) C satisfies the axioms of the modal system S5 (see Proposition 1).

d) The *induction* axiom is valid:

$$\text{INDUCTION AXIOM:} \qquad C(\varphi \supset E\varphi) \supset (\varphi \supset C\varphi).$$

e) If $\varphi \supset E\varphi$ is valid in S then $\varphi \supset C\varphi$ is valid in S.

f) The *fixpoint* axiom is valid:

$$\text{FIXPOINT AXIOM:} \qquad C\varphi \supset \varphi \wedge EC\varphi.$$

Proof: (a) follows by a straightforward induction on m showing that $(S, \rho, k) \models E^m\varphi$ iff $(S, \rho', k) = \varphi$ for all (ρ', k) of distance $\leq m$ from (ρ, k). Part (b) follows directly from (a). The proof of part (c) is identical to the proof of Proposition 1, substituting C for K_i and $\sim$ for $\overset{i}{\leftrightarrow}$. For (d), assume that both φ and $C(\varphi \supset E\varphi)$ hold at $e = (\rho, k)$. We prove by induction on m that φ holds at all points of distance $\leq m$ from e. The case $m = 0$ follows from our initial assumption. Assume that the claim holds for m, and let e' be a point satisfying $\delta(e, e') = m + 1$. It follows that there is a point e'' such that $\delta(e, e'') = m$ and $\delta(e'', e') = 1$. By the inductive hypothesis φ holds at e''. Since $C(\varphi \supset E\varphi)$ holds at e and $e \sim e''$, part (a) implies that $\varphi \supset E\varphi$ holds at e''. It follows that $E\varphi$ holds at e'', and since $\delta(e'', e') = 1$, we have that φ holds at e'. By induction we have that φ holds at all points reachable from (i.e., similar to) e, and by part (a) we have that $C\varphi$ holds at e, and we are done. Part (e) now follows since if $\varphi \supset E\varphi$ is valid in S then by (b) $C(\varphi \supset E\varphi)$ is also valid in S, and by (d) we have that $\varphi \supset C\varphi$ is also valid in S. For part (f), the validity of $C\varphi \supset \varphi$ is immediate. By part (c) we have that C satisfies the positive introspecton axiom, and hence $C\varphi \supset CC\varphi$ is valid. By definition of $C\psi$ we have that $C\psi \supset E\psi$ is valid, and taking $\psi = C\varphi$, we thus have that $C\varphi \supset CC\varphi \supset EC\varphi$ is valid, and we are done. $\bowtie$

It is interesting to note that in contrast to the case of implicit knowledge, the basic properties of E and C (which we have defined here relative to the set of active processors) are the same as those of $E_{\cdot}$ and $C_{\cdot}$, stated in [HM]. In particular, C satisfies all of the axioms of the logical system S5 (cf. [HM2]), not only the axioms mentioned above.

Proposition 7 is very useful in relating common knowledge and actions that are guaranteed to be performed simultaneously. For example, we can use Proposition 7(b) and 7(e) 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 8: 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) $(S, \rho, k) \models C(\text{"All processors are deciding } v\text{"})$, and

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

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. Thus, by Proposition 7(e) we have that $\varphi \supset C\varphi$ is valid in S, 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 Proposition 7(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. $\bowtie$

The reasoning used in proof of Corollary 8 is typical of the way Proposition 7(a) and (b) together with the consequence closure and induction axioms are used to prove that certain facts are common knowledge. We will use such reasoning again in later proofs.

3. Analysis of a simple protocol

In this section we take a close look at t-uniform systems $S_{\mathcal{F}}$ in which all processors follow a simple and fairly general protocol $\mathcal{F}$:

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.

This protocol was named the *maximal information protocol* by Hadzilacos (cf. [H]). We are interested in determining what facts about the run become common knowledge at the different stages of the execution of this protocol. Intuitively, the protocol $\mathcal{F}$ should provide the processors with "as much knowledge as possible" about the initial configuration and the pattern of failures, and should facilitate the ability of the system to perform actions that depend on the initial configuration. One of the relevant properties of this protocol is that every processor is required to send messages to all other processors in every round. This ensures among other things that tha failure of a processor will be known to all processors at most one rounds after the round in which the processor fails.

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 the processor's view, and an active processor's view grows monotonically with time, it is the case that if φ is stable then (as long as at least one processor remains active) so are $E\varphi$ and $C\varphi$. As we have seen, $I\varphi$ is not necessarily stable.

A round in which no processor fails is called a *clean* round. A round that is not clean is called *dirty*. If no processor that fails in round k fails to send to a processor that is active at time k, then round k is said to be *seemingly clean*. Notice that a clean round is

in particular seemingly clean. Individual processors cannot, in general, determine whether a round is clean, seemingly clean, etc. Roughly speaking, if, for some k, round k of a run in which the processors all follow $\mathcal{F}$ is clean, then every active processor's view at the end of round k includes the view of all of the active processors at time $k-1$. In particular this implies 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 5 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. The above discussion is made precise by the following theorem:

Theorem 9: Assume that $t \leq n-1$.

a) Let φ be a stable fact such that $(S_{\mathcal{F}}, \rho, k-1) \models I\varphi$.
If round k of ρ is seemingly clean then $(S_{\mathcal{F}}, \rho, k) \models E\varphi$.

b) Let φ be a fact about the initial configuration.
If $(S_{\mathcal{F}}, \rho, \ell) \models C$("a seemingly clean round has occurred") then
$(S_{\mathcal{F}}, \rho, \ell) = I\varphi$ iff $(S_{\mathcal{F}}, \rho, \ell) \models C\varphi$.

Proof: By definition, $(S_{\mathcal{F}}, \rho, k-1) \models I\varphi$ iff $(S_{\mathcal{F}}, \rho, k-1) \models I_G\varphi$ for $G = A(\rho, k-1)$. If round k is seemingly clean then all processors active at time k receive round k messages from all of the processors in G, and hence the view of each active processor at time k has a copy of $v(G, \rho, k-1)$, and it follows that every active processor at time k knows φ. For part (b), let φ be a fact about the initial configuration and let ψ be the fact *"a seemingly clean round has occured"*. Let (ρ', ℓ) be an execution satisfying $(S_{\mathcal{F}}, \rho', \ell) \models \psi$. By Theorem 5, if $(S_{\mathcal{F}}, \rho', \ell) \models I\varphi$ then $(S_{\mathcal{F}}, \rho', k) \models I\varphi$ for all $k \leq \ell$. Given that ψ holds at (ρ', k), let round k of ρ' be a seemingly clean round, where $k \leq \ell$. Since $(S_{\mathcal{F}}, \rho', k-1) \models I\varphi$, by part (a) we have that $(S_{\mathcal{F}}, \rho', k) \models E\varphi$. $E\varphi$ is stable because φ is, and therefore $(S_{\mathcal{F}}, \rho, \ell) \models E\varphi$. By Corollary 6 we have that $(S_{\mathcal{F}}, \rho', \ell) \models E(I\varphi)$. We have just shown that $\psi \supset (I\varphi \supset E(I\varphi))$ is valid in $S_{\mathcal{F}}$. Thus, by Proposition 7(b) we have that $C(\psi \supset (I\varphi \supset E(I\varphi)))$ is also valid in $S_{\mathcal{F}}$. Now assume that ρ is a run satisfying $(\rho, \ell) \models C\psi$. By the consequence closure axiom for C (Proposition 7(c)), we have that $(S_{\mathcal{F}}, \rho, \ell) \models C(I\varphi \supset E(I\varphi))$. And by the induction axiom we have that $(S_{\mathcal{F}}, \rho, \ell) \models I\varphi \supset C(I\varphi)$. Since $C(I\varphi) \supset C\varphi$ is valid, we also have that $(S_{\mathcal{F}}, \rho, \ell) \models I\varphi \supset C\varphi$. Finally, since $C\varphi \supset I\varphi$ is valid, we have that $(S_{\mathcal{F}}, \rho, \ell) \models I\varphi \equiv C\varphi$, and we are done. $\bowtie$

As a corollary of Theorem 9 we can now show:

Corollary 10: Let φ be a fact about the initial configuration.

a) $(S_{\mathcal{F}}, \rho, t-1) \models I\varphi$ iff $(S_{\mathcal{F}}, \rho, t+1) \models C\varphi$.

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

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. All runs of $S_{\mathcal{F}}$ have the property that no more than t processors fail during the run. Given that a processor falure occurs in a

unique round, we have that one of the first $t+1$ rounds of every run of $S_{\mathcal{F}}$ must be clean. Since a clean round is in particular seemingly clean, Proposition 7(b) implies that at time $t+1$ it is common knowledge in all runs of $S_{\mathcal{F}}$ that a seemingly clean round has occured. Part (a) now follows from Theorem 9(b). For the proof of part (b), we need a slightly stronger variant of Theorem 9(b), which states that if it is common knowledge that there has either been a clean round or that there is at most one processor then $I\varphi$ holds iff $C\varphi$ does. The proof of this fact is completely analogous to that of Theorem 9(b), given that $I\varphi \equiv C\varphi$ is trivially true when there is at most one active processor. $\ltimes$

As a consequence of Theorem 9 and Corollary 10 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 well-known 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 $\mathcal{F}$, 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 Proposition 7(e) 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 $S_{\mathcal{F}}$ 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 time 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 $D(\rho, k)$, and let $d(k)$ be a variable whose value at any point in ρ is $d(\rho, k)$. By Theorem 9(b) we have 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. We are about to show that the protocol $\mathcal{F}$ guarantees that if it ever becomes implicit knowledge that $D \geq j$ then at time $t + 1 - j$ it is common knowledge that $D \geq j$ (and, therfore, that a clean round has occurred). 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. Finally, we define the *wastefulness* of a run ρ, denoted $\mathcal{W}(S, \rho)$, by:

$$\mathcal{W}(S, \rho) \stackrel{\text{def}}{=} \max_{\ell \geq 0} \mathcal{W}(S, \rho, \ell).$$

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 $S_{\mathcal{F}}$:

Lemma 11: Let $t \leq n - 1$.

a) If $(S_{\mathcal{F}}, \rho, \ell) \models I(D \geq j)$ then $(S_{\mathcal{F}}, \rho, \ell) \models I(d(k) \geq j)$ for some $k \leq \ell$.

b) If $I(d(k) \geq j)$ holds at time k then either $E(d(k) \geq j)$ or $I(d(k-1) \geq j)$ holds at time $k - 1$

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

Proof: For part (a), let $\rho \in S_{\mathcal{F}}$ satisfy $(S_{\mathcal{F}}, \rho, \ell) \models I(D \geq j)$, and assume that for no k is it the case that $(S_{\mathcal{F}}, \rho, \ell) \models I(d(k) \geq j))$. Let ρ' be a run of $\mathcal{F}$ such that $(\rho', 0) = (\rho, 0)$, and in which the only messages not delivered are those that are implicitly known at (ρ, ℓ) not to have been delivered. It is easy to check that $\rho' \in S_{\mathcal{F}}$, since no more than t processors fail in ρ'. 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 $(S_{\mathcal{F}}, \rho, \ell) \models I(D \geq j)$. We now prove that $A(\rho, \ell)$ has the same view in (ρ, ℓ) and in (ρ', ℓ). This is done by showing by induction on k that the set of processors that are implicitly known at (ρ, ℓ) to have been active at time $k \leq \ell$ have the same views at time k in both ρ and ρ'. 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 sets $G(k)$ are the sets of processors implicitly known at (ρ,ℓ) to have been active at time k. The sets $g(p_i,k-1)$ are the sets of processors that send a message to p_i in round k. By requiring messages to contain the sender's complete view, the protocol $\mathcal{F}$ guarantees that a processor is implicitly known at (ρ,ℓ) to have been active at time k iff the processor's view at (ρ,k) is implicitly known. Thus, 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; we prove it for $k+1$. Let $p_i \in G(k+1)$. Observe that p_i's view at $(\rho,k+1)$ is determined by its view at (ρ,k) and by the view of the group $g(p_i,k)$ at (ρ,k). Since by the inductive hypothesis we have that $g(p_i,k) = g'(p_i,k)$, and that $v(g(p_i,k),\rho,k) = v(g'(p_i,\rho',k)$, and that $v(p_i,\rho,k) = v(p_i,\rho',k)$, it follows that $v(p_i,\rho,k+1) = v(p_i,\rho',k+1)$. 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 $(S_{\mathcal{F}},\rho,k) \models I(d(k) \geq j)$. 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, by requiring all processors to send messages to all of the other processors in every round, $\mathcal{F}$ 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 part (c), assume that $\mathcal{W}(\rho,k) = j$. Then by part (a) there is some $k' \leq k$ such that $(S_{\mathcal{F}},\rho,k) \models I(d(k') \geq j)$. 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. $\bowtie$

We now have:

Theorem 12: Let $t \leq n-1$.

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

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

Proof: The "if" direction of part (a) is immediate from the fact that $C\varphi \supset \varphi$ is valid. We now show the other direction. Assume that $\mathcal{W}(S_{\mathcal{F}}, \rho) \geq j$. Then for some $\ell \geq 0$ it must be the case that $\mathcal{W}(S_{\mathcal{F}}, \rho, \ell) \geq j$, and hence $(S_{\mathcal{F}}, \rho, \ell) \models I(D \geq j)$. By Lemma 11(a) there is some $k \leq \ell$ for which $(S_{\mathcal{F}}, \rho, \ell) \models I(d(k) \geq j)$. Let k' be the largest such k. Since $d(k') \geq j$ is a fact about the first k' rounds, we have by Theorem 5 that $(S_{\mathcal{F}}, \rho, k') \models I(d(k') \geq j)$. Since $d(k') \geq j$ implies that at least $k' + j$ processors must have failed by time k', we have that $k' \leq t - j$. Furthermore, $(S_{\mathcal{F}}, \rho, k'+1) \not\models I(d(k'+1) \geq j)$ implies that no new processor failure becomes visible to the active processors in round $k'+1$, and thus in particular round $k'+1$ must be seemingly clean. Since "$d(k') \geq j$" is a stable fact, it follows from Theorem 9(a) that $(S_{\mathcal{F}}, \rho, k'+1) \models E(d(k') \geq j)$, and hence that $(S_{\mathcal{F}}, \rho, \ell) \models E(d(k') \geq j)$ for all $\ell \geq k'+1$. In particular, since $t+1-j \geq k'+1$, we have that $(S_{\mathcal{F}}, \rho, t+1-j) \models E(d(k') \geq j)$. Let ψ be the fact "$\mathcal{W}(S_{\mathcal{F}}$, "the current run") $\geq j$. By Corollary 6 we have that $E(d(k') \geq j) \supset E(I(d(k') \geq j))$, and since $(d(k') \geq j) \supset \psi$ is valid, we also have that $(S_{\mathcal{F}}, \rho, t+1-j) \models E\psi$. It follows that $(S_{\mathcal{F}}, \rho', t+1-j) \models \psi \supset E\psi$ for all runs $\rho' \in S_{\mathcal{F}}$. Given that $t \leq n$, the only executions that are similar to an execution $(\rho', t+1-j)$ are of the form $(\rho'', t+1-j)$. Thus, by Proposition 7(a) we have that $(S_{\mathcal{F}}, \rho', t+1-j) \models C(\psi \supset E\psi)$ for all $\rho' \in S_{\mathcal{F}}$, and the induction axiom implies that all executions $(\rho, t+1-j)$ satisfy $\psi \supset C\psi$, which is the claim of part (a). For part (b), recall from the proof of part (a) that if $D \geq j$ then there must be a clean round by time $t-1-j$. By part (a), if $\mathcal{W}(S_{\mathcal{F}}, \rho) = j$ then at time $t+1-j$ it is common knowledge that $I(D \geq j)$ and therefore in particular that $D \geq j$. It follows that at time $t-1-j$ it is common knowledge that a clean round (and hence also a seemingly clean round) has occurred. The claim now follows from Theorem 9(b). $\bowtie$

Thus, certain patterns of failures help the processors to reach common knowledge of an identical view of the initial configuration early. In particular, if the wastefulness of the run is j, then the active processors obtain common knowledge of a common view of the initial configuration at time $t+1-j$. We now make precise our heretofore informal claim that it is the pattern of failures that determines the wastefulnes of the runs of $S_{\mathcal{F}}$. Given a system S, a fact φ is said to be *about the failure pattern* $(S, \rho, k) \models \varphi$ iff $(S, \rho', k') \models \varphi$ for all runs $\rho, \rho' \in S$ that have the same failure pattern. Observe that $d(k)$ and D are facts about the failure pattern by this definition. We can now show:

Lemma 13: Let φ be a fact about the failure pattern. Let σ and σ' be initial configurations, let π be a failure pattern, and let $\rho = \mathcal{F}(\sigma, \pi)$ and $\rho' = \mathcal{F}(\sigma', \pi)$. Then $(S_{\mathcal{F}}, \rho, \ell) \models I\varphi$ iff $(S_{\mathcal{F}}, \rho', \ell) \models I\varphi$, for all $\ell \geq 0$.

Sketch of proof: Assume that $(S_{\mathcal{F}}, \rho', k) \not\models I\varphi$, and let $G = A(\rho', k)$. It follows that there is a run ρ'' such that $v(G, \rho', k) = v(G, \rho'', k)$, and $(S_{\mathcal{F}}, \rho'', k) \not\models \varphi$. Let Q be the set of processors on whose initial states σ and σ' disagree. Clearly $v(G, \rho', k)$ contains the view at time 0 (i.e., initial state) of none of the processors in Q. Thus, without loss of generality $\rho'' = \mathcal{F}(\sigma', \pi'')$ for some π''. An inductive argument along the lines of the proof of Lemma 11(a) will now show that $v(G, \rho, k) = v(G, \mathcal{F}(\sigma, \pi''), \|)$. (Note that $A(\rho, k) = A(\rho', k) = G$.) But because φ is a fact about the failure pattern, it follows that $(S_{\mathcal{F}}, \mathcal{F}(\sigma, \pi''), \|) \not\models \varphi$, and hence $(S_{\mathcal{F}}, \rho, k) \not\models I\varphi$, and we are done with one direction. The other direction of the argument is symmetric. $\bowtie$

We can now define the *wastefulness* of a failure pattern π, denoted $w(\pi)$, to be $\mathcal{W}(S_\mathcal{F}, \rho)$ for a run ρ of the form $\rho = \mathcal{F}(\sigma, \pi)$ for some σ. Lemma 13 implies that $w(\pi)$ is independent of the initial configuration σ chosen, and therefore $w(\pi)$ is well-defined. Theorem 12 can now be read to state that if the failure pattern of a run is π, then at time $t + 1 - w(\pi)$ the active processors have common knowledge of a common view of the initial configuration. A closer inspection of the proofs of Theorem 5(c) and of Theorem 12 actually shows that if $w(\pi) = j$ the at time $t + 1 - j$ there is a particular k' such that the active processors all know that $d(k') \geq j$, and for no $\ell > k'$ is it the case that an active processor knows that $d(\ell) \geq j$. By Theorem 12(a), $w(\pi) = j$ iff "$w = j$" is common knowledge at time $t - 1 - j$. It follows that the identity of this number k' is also *common knowledge* at time $t - 1 - j$. Consequently, the active processors obtain common knowledge of a common view of the first k' rounds of the run, and not only of the initial configuration. Furthermore, since k' is determined by the implicitly known values of $d(k)$, Lemma 13 implies that the value of k' is uniquely determined by π.

One of the consequences of Theorem 12 and Lemma 13 is:

Corollary 14: There is a t-resilient protocol for SBA that reaches SBA in $t - 1 - w(\pi)$ rounds in all runs of the protocol in which the failure pattern is π, for all failure patterns π in which and at most t processors fail.

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

for $\ell \geq 0$ *perform the following at time* ℓ*:*
  if $K(D \geq t + 1 - \ell)$
    then *halt (and send no messages in the following rounds);*
      decide 0 if K("some initial value x_j was 0");
      decide 1 otherwise.
    else *send the current view to all processors in round* $\ell + 1$.

The K in the text of the protocol means "the processor knows", i.e., it is K_i in p_i's copy of the protocol. By Theorem 12(a) all correct processors halt after $t + 1 - \mathcal{W}(S_\mathcal{F}, \rho)$ rounds. By Theorem 12(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. $\bowtie$

The above protocol is not a protocol in the traditional sense of the word, but rather a *knowledge-based protocol*, to use the terminology of Halpern and Fagin in [HF]: a processor's actions at any given point are determined by the processor's knowledge. As they point out, not every knowledge-based protocol can be implemented. However, if the only knowledge required in the protocol is knowledge about the past, it is implementable. Thus, the above protocol can be directly translated into a standard protocol.

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 14 is optimal in the sense that for any given pattern of failures, it attains SBA no later than any other protocol for SBA does.

Corollary 8 and Theorem 12 imply that the stopping condition $K(D \geq t+1-\ell)$ implies $C(D \geq t+1-\ell)$. In fact, we will be able to show that this protocol is equivalent to the following protocol:

for $\ell \geq 0$ *perform the following at time* ℓ:
  if C("some initial value was 0")
    then *decide 0 and halt*
    else if C("some initial value was 1")
      then *decide 1 and halt*
      else *send the current view to all processors in round* $\ell+1$.

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 protocols 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. We first prove a lemma that is necessary for our proof of this fact. Roughly speaking, this lemma says that if $D(\rho,\ell) \leq t-\ell$ and p is the last processor to fail in ρ, then (ρ,ℓ) is similar to an execution in which p doesn't fail, and all other processors behave as they do in ρ. To make this precise we make the following definition: Given a failure pattern π, the failure pattern π^{-p} is defined to be $\pi - \langle p, k(p), Q(p)\rangle$ if there is a triple of the form $\langle p, k(p), Q(p)\rangle$ in π, and to be π if p is not designated to fail according to π. Given a run $\rho = \mathcal{P}(\sigma,\pi)$, we define ρ^{-p} to be $\mathcal{P}(\sigma,\pi^{-p})$. If $\mathcal{P}$ does not require all processors to send messages to all other processors in every round, ρ can be said to display a number of failure patterns π. However, it is easy to check that if $\mathcal{P}(\sigma,\pi) = \mathcal{P}(\sigma,\pi')$ then $\mathcal{P}(\sigma,\pi^{-p}) = \mathcal{P}(\sigma,\pi'^{-p})$, so that ρ^{-p} is well defined. We can now show:

Lemma 15: Let $t \leq n-2$, and let S be a t-uniform system for $\mathcal{P}$, with $\rho = \mathcal{P}(\sigma,\pi) \in S$. If $D(\rho,\ell) \leq t-\ell$ and no processor fails in ρ in a later round than p does, then $(\rho,\ell) \sim (\rho^{-p},\ell)$.

Proof: If p does not fail in ρ then $\rho = \rho^{-p}$, and the claim trivially holds. Thus, let k be the round in which p fails in ρ, and notice that by assumption no processor fails in ρ at a later round. If $k > \ell$ then $(\rho,\ell) = (\rho^{-p},\ell)$ and thus clearly $(\rho,\ell) \sim (\rho^{-p},\ell)$. We still need to show the claim for $k \leq \ell$. We do this by induction on $j = \ell - k$.

Case $j = 0$ (i.e., $k = \ell$): Let $q_i \neq q_j \in A(\rho,\ell)$ be two processors active at (ρ,ℓ). Such processors exist by the assumption that $t \leq n-2$. Clearly, q_i's view at (ρ,ℓ) is independent of whether or not p sent a message to q_j 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_j in round ℓ of (ρ',ℓ). (If p sends q_j a message in round ℓ of ρ, then $\rho = \rho'$.) Now, since p does send q_j a message in round ℓ of (ρ',ℓ), processor q_j's view at (ρ',ℓ) is independent of whether p fails in (ρ',ℓ), (it is

consistent with q_j's view at (ρ', ℓ) that p sends messages to all processors in round ℓ, and thus $(\rho', \ell) \sim (\rho^{-p}, \ell)$. By transitivity of $\sim$ we also have that $(\rho, \ell) \sim (\rho^{-p}, \ell)$.

Case $j > 0$ (i.e., $k < \ell$): Assume inductively that the claim holds for $j - 1$. 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 $(\rho', \ell) \sim (\rho^{-p}, \ell)$. By transitivity of $\sim$ we have that $(\rho, \ell) \sim (\rho^{-p}, \ell)$. Now assume that $s > 0$ and that the claim is true for $s - 1$. Let ρ_s be a run 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, $\rho = \rho_s^{-q_s}$, and 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 $\rho' = \rho_s'^{-q_s}$. 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 $(\rho', \ell) \sim (\rho^{-p}, \ell)$. By the symmetry and transitivity of $\sim$, we have that $(\rho, \ell) \sim (\rho^{-p}, \ell)$, and we are done. $\bowtie$

The proof of Lemma 15 is a generalization and simplification of the basic inductive argument in the lower bound proofs of [DS], [LF], and [CD]. Notice that the run ρ^{-p} in the statement of Lemma 15 has the following properties: (i) if ρ is not free of failures, then the number of processors that fail in ρ^{-p} is one fewer than in ρ; (ii) $D(\rho^{-p}, \ell) \leq t - \ell$, and (iii) $(\rho^{-p}, 0) = (\rho, 0)$. We can now use Lemma 15 to show:

Theorem 16: 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 15 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) does not determine whether the initial state of q_s in ρ_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 15 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 in 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 15, 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). $\bowtie$

Remarks: (a) The assumption of independence of the set of initial configurations is essential to the lower bound in Theorem 16. Lemma 15 is independent of this assumption. In fact, Lemma 15 can also be used to characterize non-independent systems. E.g., in systems in which it is guaranteed that processors p_1 and p_2 have an identical initial state, their initial state will become common knowledge at time t at the latest. Details are left to the reader.
(b) Lemma 15 and Theorem 16(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]). Whereas the crash failure model is weaker (i.e., is subsumed by) most other models of failures, a further weakening of this model is to assume that the processors send their messages to other processors in a particular order, and an initial segment of the messages sent by a failing processor in its round of failure are delivered (cf. [CD]). Without loss of generality we may assume that the protocol a processor follows determines this order as it determines all other actions the processor performs. The proof of Lemma 15 goes through for this model also. The only detail that must be added to the proof is that the processor $q_s \in Q$ should be the last processor (among those in Q) to whom p sends a message in round k. Details are left to the reader.

As we will see in the sequel, Theorem 16(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, Proposition 7(a) and Theorem 16(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 17: Let $t \leq n-2$, and let S be an independent t-uniform system. If $(S, \rho', \ell) \not\models \varphi$ holds for some $\rho' \in S$ satisfying $\mathcal{W}(S, \rho') \leq t - \ell$, then $(S, \rho, \ell) \not\models C\varphi$ for all $\rho \in S$ satisfying $\mathcal{W}(S, \rho) \leq t - \ell$. $\bowtie$

Theorem 17 and Theorem 12(b) completely characterize when non-trivial facts about the initial configuration become common knowledge in the runs of $S_{\mathcal{F}}$. In a precise sense, they imply that the only fact that is common knowledge at (ρ, k), for $k \leq t - \mathcal{W}(S_{\mathcal{F}}, \rho)$, is that the wastefulness is less than $t + 1 - k$. Formally, we have:

Corollary 18: Let $t \leq n-2$, let $S_{\mathcal{F}}$ be an independent t-uniform system for $\mathcal{F}$, and let $\mathcal{W}(S_{\mathcal{F}}, \rho) \leq t - \ell$. Then $(S_{\mathcal{F}}, \rho, \ell) \models C\varphi$ iff for all $\rho' \in S_{\mathcal{F}}$ such that $\mathcal{W}(S_{\mathcal{F}}, \rho', \ell) \leq t - \ell$ it is the case that $(S_{\mathcal{F}}, \rho', \ell) \models \varphi$. $\bowtie$

Furthermore, Corollary 8 and Theorem 17 immediately imply:

Corollary 19: Let $t \leq n-2$, let $\mathcal{P}$ be a t-resilient protocol for SBA, and let S be a t-uniform system for $\mathcal{P}$, with $\rho \in S$. Then SBA is not attained in ρ in fewer than $t+1-\mathcal{W}(S,\rho)$ rounds. $\bowtie$

Corollary 19 proves that SBA cannot be attained in the runs of $\mathcal{F}$ any earlier than it is attained by the protocol of Corollary 14. However, it still seems possible that using another protocol SBA will be attainable in fewer rounds than in the protocol of Corollary 14. We now show that this protocol is optimal in a rather strong sense: for any given initial configuration and failure pattern, no protocol attains SBA in fewer rounds than the protocol of Corollary 14. This fact follows from the following theorem, which states that the wastefulness of a run resulting from a given initial configuration and failure pattern is no greater than its wastefulness in $S_{\mathcal{F}}$. Given Corollary 19, this will imply that the protocol of Corollary 14 always attains SBA at the earliest possible time, given the initial configuration and failure pattern.

Theorem 20: Let S be a t-uniform system for a protocol $\mathcal{P}$, let $\rho = \mathcal{P}(\sigma,\pi)$, and let $\hat{\rho} = \mathcal{F}(\sigma,\pi)$. Then $\mathcal{W}(S,\rho) \leq \mathcal{W}(S_{\mathcal{F}},\hat{\rho})$.

Proof: We will show a more general fact from which the theorem will follow. Given an initial configuration σ', and a failure pattern π', let $\rho' = \mathcal{P}(\sigma',\pi')$ and $\hat{\rho}' = \mathcal{F}(\sigma',\pi')$. 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 $\mathcal{P}$ 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 $\mathcal{W}(S_{\mathcal{F}},\hat{\rho}) < j$. Then there is a time k such that $(S,\rho,k) \models I(D \geq j)$, and $(S_{\mathcal{F}},\hat{\rho},k) \not\models I(D \geq j)$. Let $G = A(\hat{\rho},k)$ (notice that $G = A(\rho,k)$ as well). It follows that there is a run $\hat{\rho}' \in S_{\mathcal{F}}$ such that $v(G,\hat{\rho},k) = v(G,\hat{\rho}',k)$ and $D(\hat{\rho}',k) < j$. Let σ' and π' be the initial configuration and failure pattern in $\hat{\rho}'$. Let ρ' be $\mathcal{P}(\sigma',\pi')$. 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. $\bowtie$

Theorem 20 and Corollary 19 now imply that the protocol of Corollary 14 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 20, we can talk about the inherent *wastefulness* $w(\pi)$ of a failure pattern π, defined to be $\mathcal{W}(S_{\mathcal{F}}, \mathcal{F}(\sigma, \pi))$. That $w(\pi)$ is well defined follows from the fact that runs ρ of $S_{\mathcal{F}}$ have the property that $\mathcal{W}(S_{\mathcal{F}}, \rho, k)$ depends only on the pattern of failures and is independent of the initial configuration. This can be proved by a somewhat tedious but straightforward induction on k, and is left to the reader. Theorem 16 through Corollary 19 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 14 and 19 tell us that exactly $t+1-w(\pi)$ rounds are necessary and sufficient 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 can 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 $S_{\mathcal{F}}$ 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. As we have remarked in the previous section, the proofs of Lemma 11 and Theorem 12 can be used to show that at time $t+1-w(\pi)$ in a run of $S_{\mathcal{F}}$ the active processors do not attain common knowledge only of the fact that they have a identical view of initial configuration, Rather, there is a natural number $k \geq 0$ such that at time $t+1-w(\pi)$ they attain common knowledge of an identical view of the state of the system at time k. We denote this number k by $k_1(\pi)$. There is some number, say f_1, of processors that are commonly known at time $t+1-w(\pi)$ to have failed by time $k_1(\pi)$. Let $t_1 = t - f_1$. Roughly speaking, time $k_1(\pi)+1$ can now be regarded as the start of a new run, and for appropriate definitions of $d_1(k)$ and $w_1(\pi)$, we get that at time $(k_1(\pi)+1)+t_1+1-w_1(\pi)$ the system will attain common knowledge of a common view of the state of the system at time $k_1(\pi)+1$. Interestingly, it can be shown that $(k_1(\pi)+1)+t_1+1-w_1(\pi) = t+2-w(\pi)$. That is, one round after the processors attain common knowledge of (a common view of) the state of the run at time $k_1(\pi)$, they attain common knowledge of a common view of the state of the run at $k_1(\pi)+1$. In fact, again we have some number $k'' \geq 0$ such that the processors have common knowledge at time $t+2-w(\pi)$ of a common view of the state of the system at time k''. Denoting this number by k_2, the above analysis can be repeated. We leave further details to the interested reader.

The result of the analysis discussed in the preceding paragraph is that at any point after time $t+w(\pi)$ in a run of $\mathcal{F}$ the active processors have common knowledge of a common view of the first k rounds, for a number k that can be computed given the failure pattern π. Following every round after time $t+1-w(\pi)$ the active processors attain common knowledge of a common view of at least one additional round. Consequently, there is a *window of common plausibility* of a number of the most recent rounds about which no non-trivial facts are common knowledge, and a common view of all preceding rounds is common knowledge. The size of this window at a given point is t minus the number of

processors that (at that point) are not commonly known to have failed. This classification of what facts are common knowledge in the runs of S_τ provide good upper bounds on when a simultaneous action that depends on the first k rounds can then be carried out by all active processors in a consistent way. The lower bound results of this section can imply that these bounds are tight *in all runs*, and thus we have a complete characterization of when simultaneous actions that depend on the first k rounds can be carried out, as a function of the failure pattern.

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. We now summarize our investigation of SBA. Every failure pattern π can be ascribed an inherent *waste* $w(\pi)$ such that $0 \leq w(\pi) \leq t - 1$, with the property that no protocol for SBA can reach SBA in less than $t + 1 - w(\pi)$ in a run that displays the failure pattern π. Furthermore, we have provided a protocol that guarantees to always reach SBA in exactly $t+1-w(\pi)$. The analysis presented in the previous sections applies to problems other than SBA. In this section we discuss some of these applications, in order 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, 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 16(b) immediately implies that the active processors do not have common knowledge of any non-trivial fact about the run before time $t + 1 - w(\pi)$, in any run of a t-resilient protocol with failure pattern π. The WSBA requirement is a non-trivial requirement, since when the active processors decide 1 they must have common knowledge that it is not the case that all processors started with 0 and no falure occured. Thus, WSBA cannot be reached before time $t + 1 - w(\pi)$. 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 16 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 that a non-trivial action is performed simultaneously must require $t + 1$ rounds in the worst case. We 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, a t-resilient protocol for bivalent agreement is a protocol P with the property that all runs of the independent t-uniform system S for P in which the set of initial configurations is $\{0,1\}^n$ satisfy clauses (1)–(3), and at least one run of S decides 0, and at least one

run decides 1. Proposition 7 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 12(b) implies that at the end of round 2 of $S_{\mathcal{F}}$ 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 $\mathcal{F}$ 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 17 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. Ray Strong has pointed out that the above protocol can be used to achieve 2^{n-t}-valent agreement in two rounds.

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 Proposition 7 and Theorems 17 and 20 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 Corollary 10 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 $\mathcal{F}$. 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 $\mathcal{F}$ in which at most k consecutive rounds are dirty, it is common knowledge at time $k-1$ that a clean round has occurred, and $\mathcal{F}$ 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, and 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 15 and Theorem 16 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 $\mathcal{P}$ that display such behavior will be called a *visibly restrained* system for $\mathcal{P}$. It is possible to show that in the visibly restrained system for the simple protocol $\mathcal{F}$ 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 $\mathcal{F}$ 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.

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 various facts about the system become common knowledge given an upper bound of t on the number of possible faulty processors. This problem is shown to directly correspond to the question of when simultaneous actions of various types can be performed by the processors in such a system. In particular, this is a generalization of Simultaneous Byzantine Agreement and related problems. 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, Ray Strong, Mark Tuttle and Moshe Vardi for stimulating discussions.

References

[CD] B. Coan and C. Dwork, Simultaneity is harder than agreement, *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, J.Y. Halpern ed., Morgan Kaufmann, pp. 187-206.

[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 January 1986 is available as IBM research report *RJ 4421*. Early versions appeared in *Proceedings of the Third ACM Symposium on the Principles of Distributed Computing*, 1984, pp. 50–61; and 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.