Yale University
Department of Computer Science

Efficient Protocols for Attaining Common Knowledge and Simultaneous Byzantine Agreement

Ruben Michel

YALEU/DCS/TR-603
January 1988

This work was supported in part by the Office of Naval Research under Grant N00014-82-K-0154

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR-603	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) EFFICIENT PROTOCOLS FOR ATTAINING COMMON KNOWLEDGE AND SIMULTANEOUS BYZANTINE AGREEMENT		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ruben Michel		8. CONTRACT OR GRANT NUMBER(s) ONR: N00014-82-K-0154
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Computer Science Yale University 51 Prospect Street New Haven, CT 06520		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE January 1988
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research 800 N. Quincy Arlington, VA 22217 ATTN: Dr. R.B. Grafton		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distributed unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

knowledge	Byzantine failures
common knowledge	simultaneous Byzantine agreement
distributed protocols	synchronons systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Motivated by recent research in the problems of attaining Common Knowledge and Simultaneous Byzantine Agreement in the crash and omission models, we study these problems in a more malicious scenario where incorrect processors may transmit arbitrary messages.

This paper introduces the notion of common knowledge informative protocols, which are protocols that attain, in a way, maximal common knowledge. After characterizing these protocols we design a common knowledge informative protocol which is maximally communication efficient according to various natural complexity measures. (see reverse side)

20. Abstract (contd.)

This protocol allows us to derive a worst case exponential lower bound on the number of bits that correct processors transmit in runs of common knowledge informative protocols and in runs of protocols which attain an eager type of Simultaneous Byzantine Agreement.

Efficient Protocols for Attaining Common Knowledge and Simultaneous Byzantine Agreement

Ruben Michel

Abstract

Motivated by recent research in the problems of attaining Common Knowledge and Simultaneous Byzantine Agreement in the crash and omission models, we study these problems in a more malicious scenario where incorrect processors may transmit arbitrary messages.

This paper introduces the notion of common knowledge informative protocols, which are protocols that attain, in a way, maximal common knowledge. After characterizing these protocols we design a common knowledge informative protocol which is maximally communication efficient according to various natural complexity measures.

This protocol allows us to derive a worst case exponential lower bound on the number of bits that correct processors transmit in runs of common knowledge informative protocols and in runs of protocols which attain an eager type of Simultaneous Byzantine Agreement.

1 Introduction

The interest in designing protocols for distributed networks and for multiprocessor computers is a direct consequence of the increasing usage and popularity of these systems.

As observed in various recent papers, such as [DM], [FI], [HM] and [MT], the classical notion of common knowledge emerges naturally from the study of coordination and simultaneity in multiparticipant systems. Roughly speaking, a fact is *common knowledge* if it is true, the participants know it, the participants know that the participants know it, and so forth. In a way, common knowledge plays the role of a virtual shared memory in systems in which memory allocation is primarily local.

This work was supported in part by the Office of Naval Research under Grant N00014-82-K-0154

We analyze the problem of attaining common knowledge in a standard network which is fully connected, synchronous, and in which some processors are faulty. A faulty processor may transmit arbitrary messages only during a single tick of the clock – a *round.* The faulty processor is correct before that round and it does not transmit at all thereafter. This is, in effect, a single round Byzantine behavior. Our approach generalizes to more complicated failure patterns and it extends to related problems such as attaining knowledge in distributed systems and achieving Simultaneous Byzantine Agreement.

After presenting our model for the distributed system – which closely resembles the classical model in [PSL] – we introduce the notion of a *common knowledge informative protocol, ck-informative* for short. The first approach that we consider is to let a protocol be ck-informative if some basic facts become common knowledge as early as possible to processors following that protocol. This approach yields interesting results in both the crash model and the omission model (cf. [MT]). Unfortunately, it does not extend to the more malicious models (e.g. the Byzantine model) since the performance of the faulty processors in runs of different protocols cannot be readily compared. We consider therefore a slightly different approach: A protocol is ck-informative if at each round the processors following that protocol attain as much common knowledge about basic facts as they would have attained had each of them transmitted at that round all its knowledge.

The definition of a ck-informative protocol does not provide us with an intuition of how to design such protocols. To this end we introduce a basic notion, *conveying.* A processor p conveys a fact to another processor q if p is certain that q will know that fact if it trusts p. This notion of conveying allows us to state a concise characterization of ck-informative protocols.

Equipped with this characterization we develop a ck-informative protocol which we call *the New Information Protocol,* NIP. We prove that the intrinsic parameter governing NIP's performance, in terms of communication complexity, processing time and storage space, is the number of *actual lies* performed by the crashing processors. Stated briefly, for any fixed network size, NIP's complexity is linear in the number of actual lies. We prove that NIP is maximally communication efficient according to various natural complexity measures. A corollary of this section is the construction of a maximally communication efficient simulation of the standard full-view protocol (cf. [PSL]). In appendix C we develop an efficient procedure for evaluating the basic facts that are common knowledge at each round.

We are naturally led to analyze the complexity of determining common knowledge using ck-informative protocols as a function of the parameter t, the standard bound on the number of faulty processors. We prove in the Byzantine model that, for every ck-informative protocol there exists a run of that protocol in which some processor transmits at least c^t bits at a round in which it is correct, for $c > 1$. A refinement of this proof shows that the same worst case lower bound holds in the problem of achieving Simultaneous Byzantine Agreement when corresponding *sba-informative* protocols are used.

This paper is organized as follows: In section 2 we introduce our intuitive model, followed by its formalization in section 3. In section 4 we present a brief overview of the knowledge formalism used in this paper. In section 5 we introduce the concept of a ck-informative protocol. We then present the basic notion of conveying information that allows a concise characterization of the ck-informative protocols. In section 6 we develop our ck-informative protocol NIP. In addition, we prove that NIP is linear in the number of actual lies performed in the network, and that it is maximally communication efficient according to a natural complexity measure. In section 7 we analyze the communication complexity of ck-informative protocols as a function of the parameter t. We derive an exponential lower bound which we then extend to the problem of achieving Simultaneous Byzantine Agreement.

In appendix A we present a fairly precise description of NIP. In appendix B we show that the consistency test of messages performed in NIP is as effective as the most general consistency test. In appendix C we introduce the concept of the critical round and an efficient algorithm for its evaluation in runs of NIP. Finally, in appendices D through H we prove theorems 1 through 5.

2 The Intuitive Model

Let $P = \{p_1, p_2, \ldots, p_n\}$, $n \geq 2$, denote the finite set of processors in a distributed network in which every processor can communicate with every other processor using *messages*. Messages are strings over a finite alphabet Δ.

The network is *synchronous*. This means that communication progresses in discrete rounds. Each round has two phases: At the beginning of each round every correctly operating processor transmits to other processors using messages, and outputs using strings over a finite alphabet Γ. At the end of each round l, $l = 1, 2, \ldots$ every processor p receives all the messages that were transmitted to it at the beginning of that round and it also receives an *external input*, $INPUT(p,l)$, which is a string over a finite alphabet Σ. Before communication begins, i.e. at the end of round 0, the processors receive an external input $INPUT(p,0)$.

If a processor fails in any way during a round, a powerful diagnostic will detect that flaw and disable that processor's communication links at the end of that round, thereby preventing it from transmitting erroneous information in future rounds. We adopt the prevalent assumption that, for some fixed $t < n - 1$, at most t processors fail in the network.

Notice that this model is very similar to the classical model appearing in [PSL]. Moreover, since each incorrect processor may deceive only at a single round, the lower bounds that we derive in this paper are fairly strong. In the following section we present a new formalization of this model in order to make our claims rigorous.

3 Basic Definitions

In this section we formalize the intuitive model presented above.

A (transmission) protocol $\mathcal{F}$ is a set of *protocol functions* $\{F_{(p,l)}\}_{(p,l) \in P \times N}$. Each $F_{(p,l)}$ is an n-tuple of functions $(F^{p_1}_{(p,l)}, F^{p_2}_{(p,l)}, \ldots, F^{p_n}_{(p,l)})$ so that,

$$F^{p_i}_{(p,l)} : (\Sigma^*)^l \times (\Delta^*)^{n(l-1)} \rightarrow \Delta^*.$$

$F^{p_i}_{(p,l)}$ determines the message that p transmits to p_i at round l as a function of both the messages and the external inputs that p has received before l.

An output protocol $\mathcal{O}$ is a set of *output functions* $\{O_{(p,l)}\}_{(p,l) \in P \times N}$,

$$O_{(p,l)} : (\Sigma^*)^l \times (\Delta^*)^{n(l-1)} \rightarrow \Gamma^*.$$

$O_{(p,l)}$ determines the output of processor p at round l.

Let $\mathcal{N}$ denote the set of natural numbers. A *run* ρ is a tuple

$$(n, t, \mathcal{F}, \mathcal{O}, INPUT, CA, ADV)$$

where:

- $n \in \mathcal{N}$ is the number of processors.
- $t \in \mathcal{N}$ is the bound on the number of faulty processors, $t < n - 1$.
- $\mathcal{F}$ is a protocol.
- $\mathcal{O}$ is an output protocol.
- $INPUT$ is a function $P \times (\{0\} \cup \mathcal{N}) \rightarrow \Sigma^*$.
- CA, the crashing assignment, specifies which processors are faulty and at which round they crash. Formally, CA is a set of at most t pairs in $P \times \mathcal{N}$ so that no two pairs have the same first coordinate.

- ADV, the adversary, is the set of messages that each faulty processor transmits at its crashing round and its output there. Formally, it is a function that assigns to each pair in CA an element in $(\Delta^*)^n \times \Gamma^*$.

We say that ρ is a *run of* $\mathcal{F}$. Each run ρ induces a partition of $P \times \mathcal{N}$ into one of three *states*: *healthy*$[\rho]$, *ill*$[\rho]$ and *dead*$[\rho]$. Consider a pair $(p,l) \in P \times \mathcal{N}$; if for all k, $(p,k) \notin CA$, then $(p,l) \in$ healthy$[\rho]$. If, on the other hand, for some $k \in \mathcal{N}$, $(p,k) \in CA$ then:

- If $l < k$ then $(p,l) \in$ healthy$[\rho]$.
- If $l = k$ then $(p,l) \in$ ill$[\rho]$.
- If $l > k$ then $(p,l) \in$ dead$[\rho]$.

Let *failing*$[\rho] =$ ill$[\rho] \cup$ dead$[\rho]$. We will usually abuse our notation by referring to $(p,l) \in$ healthy$[\rho]$ as "p is healthy at round l in ρ". This notation extends similarly to the cases where $(p,l) \in$ ill$[\rho]$, $(p,l) \in$ dead$[\rho]$ and $(p,l) \in$ failing$[\rho]$.

A run ρ naturally induces an *execution* $EX[\rho]$, which specifies both the messages transmitted in the network and the outputs of the processors. In order to formalize this notion we first define the *view* of a processor when it is healthy at a round in some run. The view of processor p that is healthy at round l in ρ, $v[\rho](p,l)$, is the messages that p has received before and including l in ρ and its $INPUT$ there, that is, $\{INPUT(p,k)\}_{0 \leq k \leq l}$. Hereafter we restrict the domains of $F_{(p,l)}$ and $O_{(p,l)}$ to the set of views of p at $l-1$ in runs of both $\mathcal{F}$ and $\mathcal{O}$.

We define $EX[\rho]$ according to the following rules:

- CA naturally induces the state of each processor at every round in ρ.
- At the end of each round every processor p_i receives an n-tuple of messages, whose j^{th} coordinate, $1 \leq j \leq n$ $, j \neq i$, corresponds to the message p_j transmits to p_i at that round. (The i^{th} coordinate is $\emptyset$.)
- A processor that is healthy at a round transmits in $EX[\rho]$ according to $\mathcal{F}$. Assume that the messages transmitted in $EX[\rho]$ before round l were already constructed. Then the message that p transmits to p_i at l in ρ, if it is healthy there, is $F^{p_i}_{(p,l)}(v[\rho](p,l-1))$, where $p_i \neq p$.
- A processor that is healthy at a round outputs in $EX[\rho]$ according to $\mathcal{O}$. Thus, the output of p at l in ρ, if it is healthy there, is $O_{(p,l)}(v[\rho](p,l-1))$.

- A processor that is ill at a round transmits and outputs in $EX[\rho]$ according to ADV.

- A processor that is dead at a round neither transmits nor outputs in $EX[\rho]$.

Let $M[\rho](p,q,l)$ denote the message that p sends to q at round l in $EX[\rho]$. Denote by $SEG[\rho](l)$ the messages transmitted in the first l rounds of $EX[\rho]$; more precisely,

$$SEG[\rho](l) = \{(p,q,k,M[\rho](p,q,k)) \mid p,q \in P, k \leq l\}.$$

In this paper we consider only the Byzantine case (in which processors that are ill can transmit arbitrary messages) as can be implied from the definition of the adversary.

4 Knowledge Formalism

In this section we present a knowledge formalism along the lines of [DM], modified according to our needs.

A *predicate* φ is a set of runs. A predicate φ *holds at a run* ρ, denoted by $\rho \models \varphi$, if $\rho \in \varphi$. A *basic predicate* is a predicate that depends only on CA and $INPUT$.

A processor p that is healthy at round l in ρ *knows* φ, denoted by $\rho \models K_{(p,l)}\varphi$, if φ holds at all runs which are indistinguishable by p at l from ρ. To make this definition more precise consider the following equivalence relation: Two runs of the *same* protocol are (p,l)*-equivalent,* denoted by $\rho \overset{(p,l)}{\approx} \rho'$, if p is healthy at l in both and it has the same view at l in both. Thus, $\rho \models K_{(p,l)}\varphi$ iff p is healthy at l in ρ and $\rho' \models \varphi$ for all $\rho' \overset{(p,l)}{\approx} \rho$.

Let $\rho \models E_l\varphi$ denote that every processor p that is healthy at l in ρ knows φ. Let $E_l^0\varphi = \varphi$ and let $\rho \models E_l^{m+1}\varphi$ denote $\rho \models E_l(E_l^m\varphi)$ for $m \geq 0$. A predicate φ is *common knowledge* at round l in ρ, denoted by $\rho \models C_l\varphi$, if for every $m \geq 0$, $\rho \models E_l^m\varphi$.

Two runs of the same protocol are *similar at* l, denoted by $\rho \overset{l}{\sim} \rho'$, if there exist a finite sequence of runs of that protocol $\{\rho_k\}_{0<k<m}$ and a finite sequence of processors $\{p_{i_k}\}_{0<k\leq m}$ so that:

$$\rho \overset{(p_{i_1},l)}{\approx} \rho_1 \overset{(p_{i_2},l)}{\approx} \cdots \overset{(p_{i_{m-1}},l)}{\approx} \rho_{m-1} \overset{(p_{i_m},l)}{\approx} \rho'.$$

It is apparent that $\overset{l}{\sim}$ is an equivalence relation. The following basic fact, see, e.g., [CM], [DM] and [FI], establishes a clear connection between knowledge and distributed systems:

Fact 1

$$\rho \models C_l\varphi \;\; \textit{iff} \;\; \tilde{\rho} \models \varphi \;\; \textit{for all} \;\; \tilde{\rho} \;\; \textit{satisfying} \;\; \tilde{\rho} \overset{l}{\sim} \rho.$$

5 Common Knowledge Informative Protocols

We begin this section by introducing a class of protocols that attain, in some sense, maximal common knowledge at each round. These protocols are called *common knowledge informative protocols* or *ck-informative* for short.

The first approach that we consider is to let a protocol be ck-informative if some basic facts become common knowledge as early as possible to processors following that protocol. This approach yields interesting results in both the crash and the omission models (cf. [MT]). It does not extend, however, to the more malicious models, such as the Byzantine case, since the performance of the adversary in runs of different protocols cannot be readily compared.

Thus, we need a slightly different approach. Let p be healthy at round l in a run of a ck-informative protocol. We want p to maximize the common knowledge at l in the following sense: Suppose that the processors transmit at l according to some protocol functions which may be different from the ones in the ck-informative protocol. Let φ be a predicate that is common knowledge at l in that run. Then φ should also be common knowledge at l if instead p follows at l the ck-informative protocol. Since we are comparing different protocols, we restrict our attention to predicates φ that are protocol-independent, that is, basic predicates. Now, a protocol is ck-informative if every processor that is healthy at a round in a run of that protocol maximizes the common knowledge at that round.

We proceed to formalize these ideas. We first introduce a binary relation on runs of different protocols. Unfortunately, (p,l)-equivalence cannot serve that purpose, since it only relates runs of the same protocol. We say that two runs are (p,l)*-weakly-equivalent* if p is healthy at l in both, p has the same view at $l-1$ in both, and the protocol functions corresponding to processors at rounds prior to l coincide in these two runs.

Definition 1 *Two runs are* (p,l)-weakly-equivalent, *denoted by* $\rho \overset{w\text{-}(p,l)}{\approx} \rho'$ *if:*

- *p is healthy at l in both.*

- *p has the same view at $l-1$ in both.*
- *For every processor q and round k, $k < l$, the protocol functions of q at k in ρ and ρ' coincide.*

We now use this relation in order to formalize the notion of a ck-informative protocol:

Definition 2 *A protocol $\mathcal{F} = \{F_{(p,l)}\}$ is* ck-informative *if the following holds for any run ρ of that protocol: Let ρ' satisfy $\rho' \overset{w\text{-}(p,l)}{\approx} \rho$, and let φ be any basic predicate so that*

$$\rho' \models C_l\varphi.$$

Then

$$\rho'' \models C_l\varphi,$$

where ρ'' differs from ρ' only in that p transmits at l in ρ'' according to $F_{(p,l)}$.

Interestingly, this notion coincides with the notion of an optimal protocol for common knowledge appearing in [MT], when restricted to the omission model. We will see in section 7.1 that this notion bears a close relation to a problem of eagerly attaining Simultaneous Byzantine Agreement.

A natural problem that arises at this point is finding a simple characterization of the ck-informative protocols. To this end we introduce some new concepts. Consider a processor p that by checking the message that some other processor q transmitted to it at l discovered that q was ill at l. This can happen if, e.g., q sends to p at l some forged information about some other processor r, and also $M(r,p,l) \neq \emptyset$, so that p knows at l that r was healthy at $l-1$. In the Byzantine case, the only meaningful information that such a message carries about basic predicates is that q was ill at l.

This intuition motivates the following definition: The *reduced view of p at round l in ρ*, $RV[\rho](p,l)$, is the *INPUT* that p has received up to and including round l in ρ and the set of messages that p has sent and received in these rounds. We exclude, however, every message $M[\rho](q,p,k)$, $k \leq l$, for which p knew at k in ρ that q was ill at k. Such a message is replaced in the reduced view by the statement "q was detected ill at k".

We now introduce a basic notion – *conveying.* Let p be healthy at l in some run, and assume that p knows the predicate φ. Suppose that p wants to inform q at round l that the predicate φ is true. There are many ways for p to attain this goal. The simplest would be to just transmit φ. Another, probably more efficient approach, would be to let p and q have some a priori agreements so that, e.g., more typical predicates would require less communication bits than the rare ones. Conveying completely abstracts the issue of *how* information is transferred. Instead, it captures the intuition that one processor informs another one of a predicate, provided of course that the recipient trusts the sender, without mentioning at all the means whereby this is achieved, and no matter how malicious the unreliable processors are.

Definition 3 *Assume that p is healthy at l in ρ and $\rho \models K_{(p,l-1)}\varphi$.*
p conveys φ to q at l in ρ *if*

$$\rho \models K_{(p,l-1)}K_{(q,l)}\,(\textit{if } p \textit{ is healthy at } l \textit{ then } \varphi)\,.$$

We say that *p conveys its reduced view to q at l in ρ*, if p conveys to q at l in ρ the predicate $\varphi =$ "The reduced view of p at $l-1$ is $RV[\rho](p,l-1)$". A protocol is a *(reduced) conveying protocol*, $(\mathcal{R})\mathcal{CP}$ for short, if at every run of that protocol every processor conveys its (reduced) view to all the other processors at every round in which it is healthy.

In order to introduce some life into the runs that we are considering we must initiate them somehow. To this end we assume that each processor p that is healthy at round 1 conveys its $INPUT$ (at round 0) to every other processor.

Another notion we need in this section is *information symmetry.* The intuition behind this notion is that the message that processor p, which is healthy at l, transmits to a processor q at l completely determines the message that it transmits to any other processor at that round.

Definition 4 *A protocol $\mathcal{F} = \{F^q_{(p,l)}\}$ is* information symmetric *if $(F^q_{(p,l)})^{-1} \circ F^q_{(p,l)}$ is independent of q.*

To see that this definition meets the intuition that we mentioned above, consider the following rather trivial lemma:

Lemma 1 *The protocol $\mathcal{F} = \{F_{(p,l)}\}$ is information symmetric iff for every pair of runs ρ and ρ' of $\mathcal{F}$ in which p is healthy at l, for all processors q and r, $M[\rho](p,q,l) = M[\rho'](p,q,l)$ iff $M[\rho](p,r,l) = M[\rho'](p,r,l)$.*

Finally, we state the central result of this section, which is a characterization of information symmetric ck-informative protocols:

Theorem 1 *An information symmetric protocol $\mathcal{F}$ is ck-informative iff $\mathcal{F}$ is an $\mathcal{RCP}$.*

5.1 Weak Information Symmetry

In the previous section we introduced the notion of information symmetry. As we will see in this section this notion is too restrictive in several respects, therefore we will try to modify it in order to capture more naturally our intuition of information symmetry.

The changes that we want to introduce involve two aspects of information symmetry that we find inappropriately strong. Consider again processor p that is healthy at round l. Our first reservation about information symmetry is that it requires p to transmit at l to processors that it knew at $l-1$ would be dead at l. Our second reservation is that p is required to convey to q information, which p knows that q already knows, such as information that p received from q at previous rounds.

Having stated the drawbacks of information symmetry, we now develop a new notion called *weak information symmetry* that on the one hand maintains our basic intuition about information symmetry, and on the other hand avoids the drawbacks that we mentioned above.

Consider two processors q and r that p did not know at $l-1$ would be dead at l. We no longer require that the message that p sends to q at round l determine the message that p sends to r at that round. The point is that p will completely omit the messages that q transmitted to it before l from its transmissions to q at l, whereas it may convey them to r.

Not surprisingly, there is a simple solution to this problem. We want the messages that p transmits to q up to and including l *together* with the messages that q transmits to p prior to l, to completely determine the messages that p transmits to r up to and including l.

We also require that if p knew at $l-1$ that q would be dead at l, then p should not transmit anything to q at l. The skeleton of the definition of weak information symmetry is now developed.

Definition 5 *A protocol $\mathcal{F}$ is* weakly information symmetric *if it satisfies the following two properties:*

- *Let ρ and ρ' be any two runs of $\mathcal{F}$, let p be healthy at l in both runs, and assume that p did not know at $l-1$ in either run that q would be dead at l nor that r would be dead at l. Then, if $M[\rho](p,q,k) = M[\rho'](p,q,k)$ for $k \leq l$, and $M[\rho](q,p,k) = M[\rho'](q,p,k)$ for $k < l$, then also for every $k \leq l$, $M[\rho](p,r,k) = M[\rho'](p,r,k)$.*
- *For every run ρ of $\mathcal{F}$, if p is healthy at l in ρ and it knew at $l-1$ that q would be dead at l, then $M[\rho](p,q,l) = \emptyset$.*

Fortunately this weaker notion of information symmetry preserves theorem 1. In fact:

Corollary 1 *A weakly information symmetric protocol $\mathcal{F}$ is ck-informative iff $\mathcal{F}$ is an $\mathcal{RCP}$.*

Corollary 1 is the theoretical motivation for the New Information Protocol, NIP, that we develop in the next section.

6 The New Information Protocol

It is apparent from theorem 1 that the standard Full-View Protocol, FV (cf. [PSL]), in which every processor transmits its view whenever it is healthy, is ck-informative. It is well known, however, that FV is communication inefficient.

In this section we introduce another ck-informative protocol, the New Information Protocol, NIP. As indicated by its name, the basic idea behind NIP is that each processor transmits only new information at each round in which it is healthy.

NIP has four appealing properties: First, of course, it is ck-informative. Second, it is weakly information symmetric. Third, each processor transmits at every round as little information as possible, and fourth, it is maximally communication efficient under various natural complexity measures. This notion of information will not be formalized in this paper.

6.1 Message Structure in NIP

NIP resembles FV in the structure of its messages. Recall that each message in FV can be viewed as a union of *atoms*, e.g.,

$$M(p_{i_{k-1}}, p_{i_k}, l) = \bigcup \text{atom}$$

where each atom is an ordered pair of the form

$$\text{atom} = \langle \text{chain, content} \rangle.$$

Now a *chain* in this case looks like

$$p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \rightarrow p_{i_{k-2}} \overset{l-1}{\rightarrow} p_{i_{k-1}}$$

and its *content* is either an *INPUT*, the empty message $\emptyset$, or a lie. The semantics of that atom is that p_{i_1} transmitted content to p_{i_2} at $l-k+2$, and that this information reached $p_{i_{k-1}}$ at $l-1$ passing through $p_{i_3}, p_{i_4}, \ldots, p_{i_{k-2}}$. Notice, however, that some processor p_{i_f}, $1 < f < k$, might have forged that content or the head of that chain (i.e. $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \rightarrow p_{i_{f-1}}$).

We now introduce the format of the messages in NIP. This format will allow each processor to process and store its information efficiently. We begin by saying what these messages are *not*. A message that is transmitted according to NIP is *not* merely a sequence of chains queued one after the other; instead, it is formatted as a *transmission tree*. The transmission tree that p transmits to q at l, which we denote by $TT(p,q,l)$, is a tree having nodes labelled with processors' names, a root labelled p, and no two sons of any internal node having the same label. A path from a leaf to the root represents a chain that p received at $l-1$, and every such path carries the corresponding content. Notice that the root p can in fact be dispensed with. We have introduced it since it is easier to talk about transmission trees instead of transmission bushes.

Another minor difference between the messages in NIP and the messages in FV is that in NIP we allow the content of atoms to be of the form "processor q was detected ill at round l."

6.2 Transmission in NIP

In this section we develop the intuition behind NIP. There are two basic principles that make NIP an appealing protocol. The first is that each processor checks the consistency of the messages it receives, and the second is that each processor transmits only new information at each round. We now expand on these two ideas by describing their implementation in NIP.

We need a technical detail: In definition 3 we introduced the notion "p conveys the predicate φ to q at l in the run ρ", for p healthy at l. Hereafter we extend this definition also to processors p that are ill at l and to predicates φ that p claims at l to have known at $l-1$. We are assuming here that the recipient q does not detect that p was ill.

Consider a processor p that is healthy at round l in some run of NIP. At the end of that round p receives the messages that were addressed to it. The first fact that p concludes from a nonempty message that a processor, say q, transmits to it, is that q must have been healthy at $l-1$, and therefore that the information that q conveyed there is certainly trustworthy. Notice that trusting q at $l-1$ might involve trusting some other processors at $l-3$, and this in turn might involve trusting some other processors at $l-5$, and so forth. Thus the mere fact that a message is nonempty at l conveys a substantial amount of information.

The next step that p performs at l is to check the consistency of the messages it receives. This consistency check is the first basic principle in NIP. Consider the message that q transmits to p at l. The first examination that p performs on that message is a standard syntactical test. Next, for every r that p knows is healthy at $l-1$, p uses the weak information symmetry of NIP in order to check that q conveyed to it at l correct information about r. Similarly, for every r that p knows is dead at $l-1$, p checks that q conveyed to it that q did not receive a message from r at $l-1$.

We have already said what p checks in the message it received from q at l concerning each processor r that p knew was either healthy or dead at $l-1$. What can p check in that message if it did not know that r was either healthy or dead? Well, as far as p knows, r could have sent any message whatsoever to q at $l-1$. However, p *does* know that had q been healthy at l, it should have transmitted only information that was new to it, according to the second principle, and it should have checked the reliability of the information that r conveyed to it. Notice that the latter test is in effect a recursive procedure, since p might have to check next that q checked that r checked some other processor, and so forth. Now that we know what it is that p can check, we state that this is precisely what it is going to check. To test that q transmitted only information that was new to it is fairly easy. To test that q checked the consistency of the messages that were transmitted to it is somewhat more involved. p first constructs the reduced view of q at $l-1$ based on the messages that q transmitted to it up to and including l. p then checks that the messages that q claimed to have received from each such r could have passed q's examination in the reduced view that q conveyed to it.

In appendix B we prove a central result, which states that when p checks q at l regarding the message that q claimed to have received from r at $l-1$, it need only check the new information that q conveyed to p that r conveyed to q. In other words, if q conveys to p at l that r conveyed to it some information at $l-1$ that was in accordance with what q knew at $l-2$, then this type of information cannot lead to an inconsistency in the message that q sent p at l. If, on the other hand, q transmits that information rather than conveying it, then p knows immediately that q was ill, since this contradicts the principle of transmitting only new information. This lemma has a strong impact on the communication, time and space complexity of NIP. In fact it shows that NIP is *linear* in the number of lies committed in the network with respect to these three criteria.

We now discuss the second principle, namely, that each processor transmits only new information when it is healthy. Consider again processor p after it checked the consistency of the messages that it received at l. Since NIP is ck-informative, p must find a way to convey its reduced view to all the processors that might be healthy at $l+1$. Let q be a processor that transmits a non-empty message to p

at l so that p cannot determine at l that it was ill. Thus q must have conveyed to p the message that it received from every other processor, say r, which we denote by

$$M(r,q,l-1|M(q,p,l)).$$

Assume first that $M(r,p,l-1) \neq \emptyset$ and p could not determine at $l-1$ that r was ill at $l-1$. Then, the *PIVOT message of p with respect to r at* $l-1$ or, more shortly, $PIVOT(r,p,l-1)$ is $M(r,p,l-1)$. The motivation behind this name is as follows: p does *not* transmit $M(r,q,l-1|M(q,p,l))$. Instead, it transmits the atoms whereby $M(r,q,l-1|M(q,p,l))$ differs from $PIVOT(r,p,l-1)$, which we call *the new information that q conveys to p at l about r*.

What about decoding? Since $M(r,p,l-1) \neq \emptyset$ and p could not determine at $l-1$ that r was ill at $l-1$, p must have conveyed $M(r,p,l-1)$ at l. Thus, all the processors that are healthy at $l+1$ can retrieve $M(r,q,l-1|M(q,p,l))$ from the difference that p transmits at $l+1$ and the $PIVOT$.

The point in transmitting only differences is that if both q is healthy at l and r is healthy at $l-1$, then p will not have to transmit at $l+1$ any communication bit whatsoever in order to convey $M(r,q,l-1|M(q,p,l))$. If, on the other hand, either q is not healthy at l or r is not healthy at $l-1$, then each atom that p transmits at $l+1$ carries at least one lie performed by either q at l or r at $l-1$. This relation between the lies that are committed and the atoms that are transmitted plays a crucial role in the analysis of NIP.

Assume now that p managed to determine either that r was ill at $l-1$ or that the message that r transmitted to p at $l-1$ was empty. Here p will not choose $M(r,p,l-1)$ to be the $PIVOT$ message. Instead, it generates an imaginary message $M'(r,p,l-1)$ as follows: First, for every $s \neq r,p$, let

$$M'(s,r,l-2) = M(s,p,l-2)$$

up to weak information symmetry. Now let $M'(r,p,l-1)$ be the message that r would have transmitted at $l-1$ had it been healthy there and had it received the messages that we just constructed. The $PIVOT$ of p with respect to r at $l-1$ is $M'(r,p,l-1)$.

Refer to appendix A for a more detailed explanation of NIP.

6.3 The Complexity of NIP

In this section we analyze the complexity properties of NIP. More specifically, for a given processor p at round l in some run ρ of NIP, we estimate the number of bits that p transmits at l, provided, of course, that it is healthy there. We also estimate the time and the space that the routines described in appendix A use for calculating the messages that p transmits at l.

We will prove that the intrinsic parameter governing NIP's complexity in a segment is *the number of actual lies performed in that segment.* Moreover, NIP is *linear* in that parameter.

Before introducing our notion of an actual lie we make two observations: First, recall that in a ck-informative protocol each processor conveys its reduced view when it is healthy, and that its reduced view can be represented as a union of atoms. Second, consider the atom

$$\langle p_{i_1} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}, \beta \rangle,$$

that p_{i_k} conveys to q at l, where $\beta = \emptyset$ or $\beta =$ "detected ill". In NIP this atom conveys to q two basic facts, provided of course that it trusts $p_{i_2}, \ldots, p_{i_k}$: First, that $M(p_{i_1}, p_{i_2}, l-k+1) = \beta$, and second, that p_{i_1} conveys no new information to p_{i_2} at $l-k+1$ besides $M(p_{i_1}, p_{i_2}, l-k+1) = \beta$. The second fact means that p_{i_2} views the message that p_{i_1} transmits to it as the message $M'(p_{i_1}, p_{i_2}, l-k+1)$ defined at the end of section 6.2. Thus, a content can be naturally assigned to each chain of the form

$$q_{j_1} \rightarrow \ldots \rightarrow q_{j_f} \rightarrow p_{i_1} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}.$$

With these observations in mind we now introduce our notion of an actual lie. Intuitively, an actual lie is simply an atom whose content is incorrectly conveyed. More formally, let q be healthy at l in ρ, and let r be ill at l. We distinguish between two cases:

1. q does not know at l in ρ that r is ill at l and $M[\rho](r, q, l) \neq \emptyset$.

   The atom a is an *actual lie* that r conveys to q at l in ρ, if r conveys a to q at l in ρ, and

   - If $a = \langle \overset{l-1}{\rightarrow} r, \alpha \rangle$, then $INPUT[\rho](r, l-1) \neq \alpha$.
   - If
     $$a = \langle p_{i_1} \rightarrow \ldots \rightarrow p_{i_{k-1}} \rightarrow p_{i_k} \overset{l-1}{\rightarrow} r, \alpha \rangle$$
     then p_{i_k} conveys the atom
     $$\langle p_{i_1} \rightarrow \ldots \rightarrow p_{i_{k-1}} \overset{l-2}{\rightarrow} p_{i_k}, \gamma \rangle$$
     to r at $l-1$ in ρ, where $\gamma \neq \alpha$.

2. q either discovers at l in ρ that r is ill at l or $M[\rho](r,q,l) = \emptyset$.

   Let $M'(r,q,l)$ be the message whose construction we described at the end of section 6.2. a is an *actual lie* that r conveys q at l in ρ if either a is an atom whose content is incorrectly conveyed to q by r through $M'(r,q,l)$ as explained above, or a is the message $M[\rho](r,q,l)$.

This definition naturally extends to ck-informative protocols other than NIP.

Let $AL[\rho](k,l)$ denote the number of actual lies conveyed from round k to round l inclusive in the run ρ of NIP, where $k \leq l$, and for technical reasons let

$$AL^+[\rho](k,l) = AL[\rho](k,l) + 1.$$

The *content length* of an atom a, denoted by $|a|$, is the number of bits used in order to represent the content of that atom. Let

$$|\rho| = \{|a| \mid a \text{ is conveyed in } \rho\}.$$

The following theorem determines the complexity of NIP:

Theorem 2 *Let processor p be healthy at round l in the run ρ of NIP.*

1. *The number of bits that p transmits at l in ρ to another processor is less than*

   $$n((t+1)\log n + |\rho|)AL^+[\rho](l-2, l-1).$$

2. *The time needed for calculating the messages that p transmits at l using the routines described in appendix A is*

   $$cAL^+[\rho](l-3, l-1)$$

   where $c = poly(n,t,|\rho|)$. The space used in that calculation is

   $$c'AL^+[\rho](l-t-1, l-1)$$

   where, as before, $c' = poly(n,t,|\rho|)$.

6.4 Maximal Communication Efficiency of NIP

Consider the following approach for comparing the communication efficiency of different ck-informative protocols. Compare the total number of bits transmitted by the processors that are healthy in segments of two such protocols sharing some basic properties, such as identical CA and $INPUT$.

Unfortunately, this measure is inappropriate; in fact, for every given run $\sigma_{\mathcal{Q}} = (n, t, \mathcal{Q}, \mathcal{O}, INPUT, CA, ADV)$ of a ck-informative protocol $\mathcal{Q}$, there exists a ck-informative protocol $\mathcal{F}$ and an adversary ADV' such that in the run $\sigma_{\mathcal{F}} = (n, t, \mathcal{F}, \mathcal{O}, INPUT, CA, ADV')$ each processor that is healthy at a round transmits at most a single bit to every other processor, and each such processor conveys precisely the same atoms as the corresponding processor at the same round in $\sigma_{\mathcal{Q}}$. We construct the protocol $\mathcal{F}$ as follows: Say that p conveys to q at l in $\sigma_{\mathcal{Q}}$ the set of atoms $\mathcal{L}$. We build $\mathcal{F}$ so that p conveys $\mathcal{L}$ to q at l iff p transmits to q at l the bit 1. The construction of $\sigma_{\mathcal{F}}$ is clear. Thus, the communication efficiency of protocols can only be measured in some weaker sense.

For introducing our notion of communication efficiency we need the following definition. Let

$$a = \langle p_{i_1} \rightarrow \ldots \xrightarrow{k-1} p_{i_f}, \alpha \rangle.$$

The ∞-*length* of a in the run σ, $|a|_\infty$, is $\max\{|a|, 1\}$ if p_{i_f} does not convey any actual lie with chain $p_{i_1} \rightarrow \ldots \xrightarrow{k-1} p_{i_f}$ at k in σ, and otherwise, it is $\max |b|$, for atoms b with chain $p_{i_1} \rightarrow \ldots \xrightarrow{k-1} p_{i_f}$ that p_{i_f} conveys at k in σ.

A ck-informative protocol $\mathcal{CE}$ is *Communication Efficient* if for every run σ of $\mathcal{CE}$, and for every other ck-informative protocol $\mathcal{F}$, there exists a run ρ of $\mathcal{F}$ satisfying the following properties: First, the parameters n, t and CA are identical in σ and ρ. Second, the $INPUT$s in ρ are no longer than the corresponding $INPUT$s in σ. Third, for every actual lie that is conveyed in ρ there exists a comparable actual lie in σ. This means that there exists a one-to-one function mapping each actual lie $\langle ch, \alpha \rangle$ in ρ into an actual lie $\langle ch, \beta \rangle$ in σ so that

$$|\langle ch, \beta \rangle|_\infty \geq |\langle ch, \alpha \rangle|.$$

Finally, for every l, fewer bits are transmitted by the processors that are healthy in $SEG[\sigma](l)$ than by the processors that are healthy in $SEG[\rho](l)$, up to a multiplicative factor of size polynomial in n and t. The intuition here is that since exponentially (in n and t) many bits are often transmitted in runs of ck-informative protocols, the polynomial factor is quite insignificant. Moreover, this polynomial factor will allow us to establish worst case exponential lower bounds on the number of bits that processors transmit when they are healthy in runs of ck-informative protocols.

We now formalize this intuition. Let the *Communication Complexity function of* $SEG[\rho](l)$, $CC[\rho](l)$, be the number of bits transmitted by the processors that are healthy from round 1 to round l inclusive in ρ.

Let $\mathcal{Q}$ and $\mathcal{F}$ be two ck-informative protocols. Given a run σ of $\mathcal{Q}$, let *the runs of* $\mathcal{F}$ *dominated by* σ, $DOM(\mathcal{F}, \sigma)$, be the set of runs ρ of $\mathcal{F}$, so that n, t and CA

in σ and ρ coincide,

$$|\langle \xrightarrow{k} q, INPUT[\sigma](q,k)\rangle|_\infty \geq |\langle \xrightarrow{k} q, INPUT[\rho](q,k)\rangle|$$

for all processors q and $0 \leq k < l$, and there exists a one-to-one function mapping each actual lie $\langle ch, \alpha\rangle$ in ρ into an actual lie $\langle ch, \beta\rangle$ in σ so that $|\langle ch, \beta\rangle|_\infty \geq |\langle ch, \alpha\rangle|$. Notice that the parameters generating $DOM(\mathcal{F}, \sigma)$ are relatively short $INPUT$s and severely restricted adversaries.

Definition 6 *A ck-informative protocol $\mathcal{CE}$ is* communication efficient *if for some $p(n,t) = poly(n,t)$, independent of the number of actual lies performed in the network, for every run σ of $\mathcal{CE}$, ck-informative protocol $\mathcal{F}$ and round l:*

$$CC[\sigma](l) \leq p(n,t) \max_{\rho \in DOM(\mathcal{F},\sigma)} CC[\rho](l).$$

We state now the main result of this section:

Theorem 3 *NIP is a communication efficient ck-informative protocol.*

We encourage the reader to develop other notions of communication efficiency, and to prove that NIP satisfies them. Notice that NIP provides an interesting $\mathcal{O}$(actual lies) simulation of the classical FV.

7 The Complexity of ck-informative Protocols

In this section we present a lower bound on the number of bits that are transmitted by the processors that are healthy in runs of ck-informative protocols in terms of the parameter t.

Theorem 4 *For every n and t and for every ck-informative protocol with these parameters, there exists a run ρ of that protocol with $|\rho| = 1$ in which some processor transmits at least c^t bits at a round in which it is healthy, for $c > 1$.*

7.1 The Complexity of Simultaneous Byzantine Agreement

The lower bounds presented above extend to the problem of Simultaneous Byzantine Agreement, SBA (cf. [DM]). Motivated by our notion of a ck-informative protocol, we say that a protocol $\mathcal{F}$ is *sba-informative* if, roughly speaking, the correct processors transmit sufficient information so that if SBA could be achieved at a round using some other protocol functions corresponding to that round, then it would also be achieved using $\mathcal{F}$.

Definition 7 *A protocol* $\mathcal{F} = \{F_{(p,l)}\}$ *is* sba-informative *if the following holds for any run* ρ *of that protocol: Let* ρ' *satisfy* $\rho' \overset{w\text{-}(p,l)}{\approx} \rho$*, and assume that SBA is attained at* l *in* ρ'*. Then SBA is also attained at* l *in* ρ''*, where* ρ'' *differs from* ρ' *only in that* p *transmits at* l *in* ρ'' *according to* $F_{(p,l)}$.

Refining the methods above we can prove the following worst case exponential lower bound for this type of SBA, which we call *Eager SBA*.

Theorem 5 *For every* n *and* t *and for every sba-informative protocol with these parameters, there exists a run of that protocol in which some processor transmits at least* c^t *bits at a round in which it is healthy, for* $c > 1$.

Acknowledgement: I am indebted to Michael J. Fischer for stimulating discussions. I am deeply grateful to Neil Immerman. Special thanks to Joyce Gastel for substantially improving the readability of this paper. I also thank Josh Cohen Benaloh, David Greenberg, Yoav Shoham, Gregory Sullivan and Lenore Zuck.

References

[CM] Chandy K., Misra J. "How processes learn." *Distributed Computing, 1:1.*, (1986), 40–52.

[DM] Dwork C., Moses Y. "Knowledge and Common Knowledge in a Byzantine Environment I: Crash failures." *Proc. of the 1986 Conf. on Theoretical Aspects of Reasoning About Knowledge.*, Monterey, California, (March. 1986), 149–169.

[FI] Fischer M., Immerman N. "Foundations of Knowledge for Distributed Systems." *Proc. of the 1986 Conf. on Theoretical Aspects of Reasoning About Knowledge.*, Monterey, California, (March. 1986).

[HM] Halpern J., Moses Y. "Knowledge and Common Knowledge in a Distributed Environment." *Proc. of the Third ACM Symp. on the Principles of Distributed Computing.*, (1984), 50–61.

[MT] Moses Y., Tuttle M. "Programming Simultaneous Actions Using Common Knowledge." *Proc. of the 1986 Symp. on Foundations of Computer Science.*, Toronto, Canada, (Oct. 1986), 208–221.

[PSL] Pease M., Shostak R., Lamport L. "Reaching Agreement in the presence of faults." *JACM, 27:2.*, (1980), 228–234.

A The Code for NIP

In this section we present a fairly precise description of NIP. We first introduce the notations that we use in the code for NIP. Next we develop the procedures whereby the healthy processor p at round l maintains and updates its knowledge data structure, which we denote by $NK(p,l)$. Finally, we present a program that generates the messages that p transmits at round $l+1$ in runs of NIP, provided that it is healthy there. The crucial component of that program is, as expected, $NK(p,l)$.

A.1 Notations

In order to simplify our presentation we need some notations and conventions.

We adopt a PASCAL-like programming style. Procedure names are written in capital sans serif style, e.g., UPDATE_NK(p,l). Names of arrays are written in capital slanted style such as $LB(q,p,l)$. Labels are written in capital bold style, e.g. **CHECK** and comments are written in `typewriter style`.

And now some notations: $M(q,r,l-1|M(r,p,l))$ denotes the message that q transmitted to r at $l-1$ as conveyed by r to p at l. Similarly, $M(q,r,l|M(q,p,l))$ denotes the unique message that q should have transmitted to r at l had it been healthy there and had it transmitted $M(q,p,l)$ to p (recall that NIP is weakly information symmetric).

$ST(s,r,l-2|M(q,p,l))$ denotes the subtree derived from $M(q,p,l)$ (in fact from the transmission tree that it denotes), by following the path starting at the root (labelled q) going through its son labelled r and reaching the son of r labelled s. Now $ST(s,r,l-2|M(q,p,l))$ is the subtree rooted at s.

Similarly, $PRUNE(r,q,l-1|LB(q,p,l))$ denotes the subtree derived from $LB(q,p,l)$ by first following the path starting at the root (labelled q) and ending at its son labelled r. Then $PRUNE(r,q,l-1|LB(q,p,l))$ is the tree resulting from $LB(q,p,l)$ after pruning the subtree rooted at r.

A.2 The PIVOT

In section 6.2 we introduced the second principle of NIP, namely, that each processor transmits only new information when it is healthy. To this end we introduced the $PIVOT$ message.

Let p be healthy at l in a run of NIP, and let q be a processor that transmits a non-empty message to p at l so that p cannot determine at l that it was ill. Thus q must have conveyed to p the message that it received from every other processor, say r, which we denoted by $M(r,q,l-1|M(q,p,l))$.

If p could not determine at $l-1$ that r was ill at $l-1$ and $M(r,p,l-1) \neq \emptyset$, then the message that p received from r at that round is the *PIVOT* message of p with respect to r at $l-1$, $PIVOT(r,p,l-1)$.

Assume now that p managed to determine either that r was ill at $l-1$ or that the message that r transmitted to p at $l-1$ was empty. Here p will not choose $M(r,p,l-1)$ to be the *PIVOT* message. Instead, it generates an imaginary message $M'(r,p,l-1)$ as follows: First, for every $s \neq r,p$, let

$$M'(s,r,l-2) = M(s,p,l-2)$$

up to weak information symmetry. Now let $M'(r,p,l-1)$ be the message that r would have transmitted at $l-1$ had it been healthy there and had it received the messages that we just constructed. The *PIVOT* of p with respect to r at $l-1$ is $M'(r,p,l-1)$.

The intuition behind the *PIVOT* message is as follows: p does *not* transmit

$$M(r,q,l-1|M(q,p,l))$$

at $l+1$. Instead it transmits the atoms whereby $M(r,q,l-1|M(q,p,l))$ differs from $PIVOT(r,p,l-1)$. We will show in appendix A.4 how this difference is evaluated in NIP.

A.3 The nc-state

In section 3 we introduced a partition of $P \times \mathcal{N}$ based on the crashing assignment – CA. We now introduce another partition based on the knowledge that a processor has at a round in which it is healthy. More specifically, let p be healthy at l in ρ. We define the following partition of $P \times \mathcal{N}$ into five sets which we call the nc-state(p,l):

- nc-healthy$[\rho](p,l) = \{(q,k)|$ p knows at l in ρ that q was healthy at k $\}$.

  The sets nc-ill$[\rho](p,l)$ and nc-dead$[\rho](p,l)$ are defined similarly.

- nc-pseudo-healthy$[\rho](p,l) = \{(q,k)|$ p knows at l in ρ that q was either healthy or ill, but it does *not* know which of the two $\}$.

  The set nc-pseudo-dead$[\rho](p,l)$ is defined similarly. Here the uncertainty is between ill or dead rather than between healthy or ill.

Thus $(q,k) \in$ nc-pseudo-healthy$[\rho](p,l)$ means that p knows at l in ρ that q was either healthy or ill at k, but it cannot determine which of the two. This can happen in the following situation: p receives a message from q at round k, for some $k \leq l$, from which it could not determine that q was lying at k. Moreover, it does not receive any message from q after k, and it gets no additional information indicating that q was ill at k.

We will find it useful to consider the following subset of nc-ill$[\rho](p,l)$:

$$\text{nc-detected-ill}[\rho](p,l)=\{(q,k) \mid\ k \leq l \text{ and } (q,k) \in \text{nc-ill}[\rho](p,k)\ \}.$$

The idea is that if $(q,k) \in$ nc-detected-ill$[\rho](p,l)$ then it is not only true that p knows at l in ρ that q was ill at k, but it actually could determine that fact *at round k* by examining the message that q transmitted to it at k.

Also, let

$$\text{nc-failing}[\rho](p,l)= \text{nc-ill}[\rho](p,l) \cup \text{nc-pseudo-dead}[\rho](p,l) \cup \text{nc-dead}[\rho](p,l)$$

and let

$$\text{nc-known}[\rho](p,l)= \text{nc-healthy}[\rho](p,l) \cup \text{nc-dead}[\rho](p,l).$$

A.4 The Operators $\triangle$ and $\bigtriangledown$

We introduce now the two binary operators that allow the coding and decoding of the messages in NIP: $\triangle$ and $\bigtriangledown$. Consider $\triangle$ first. Denote by $C(M_i)$ the set of atoms conveyed through M_i, for $i = 1,2$. Then, $M_1 \triangle M_2$ is basically an efficient encoding of $C(M_1) \setminus C(M_2)$.

These operators might be better understood by considering a simplification of the problem. Assume for the moment that each message transmitted in NIP is really a set of atoms instead of a transmission tree. Consider the following two messages:

$$\begin{aligned} M_1 &= \{\langle ch_1, \alpha_1\rangle, \langle ch_2, \alpha_2\rangle, \langle ch_3, \alpha_3\rangle\} \\ M_2 &= \{\langle ch_1, \alpha_1\rangle, \langle ch_2, \beta_2\rangle, \langle ch_3, \alpha_3\rangle, \langle ch_4, \beta_4\rangle\} \end{aligned}$$

where ch_i, $i = 1, \ldots, 4$, denote different chains and α_i, $i = 1, \ldots, 4$, and β_i, $i = 2, 4$, denote distinct values for content. In this example

$$M_1 \triangle M_2 = \{\langle ch_2, \alpha_2\rangle, \langle ch_4, \alpha_4\rangle\}$$

where α_4 is the content corresponding to ch_4 as conveyed by the processor that transmitted M_1. Roughly speaking, the effect of $\triangle$ is to discard from its first operand each atom that also appears in its second operand (hence, it is not symmetric). Further, an atom that appears in the second operand carrying a chain *ch* that is missing from the first operand will appear in $M_1 \triangle M_2$ as an atom carrying both *ch* and the corresponding content, which was conveyed by the processor that transmitted the first operand.

More specifically, let p_{i_k} be healthy at l in a run of NIP, and let $p_{i_{k-1}}$ satisfy $(p_{i_{k-1}}, l-1) \in$ nc-pseudo-healthy$(p_{i_k}, l-1)$. Assume that p_{i_k} did not know at $l-1$ either that $p_{i_{k-2}}$ was healthy at $l-2$ or that it was dead there, i.e., $(p_{i_{k-2}}, l-2) \notin$ nc-known$(p_{i_k}, l-1)$.

We now construct the tree

$$DIF = M(p_{i_{k-2}}, p_{i_{k-1}}, l-2 | M(p_{i_{k-1}}, p_{i_k}, l-1)) \triangle PIVOT(p_{i_{k-2}}, p_{i_k}, l-2)$$

by examining the following cases.

Assume first that $M(p_{i_{k-2}}, p_{i_k}, l-2) \neq \emptyset$ and p_{i_k} did not detect at $l-2$ that $p_{i_{k-2}}$ was ill. If $p_{i_{k-1}}$ conveys to p_{i_k} at $l-1$ that $M(p_{i_{k-2}}, p_{i_{k-1}}, l-2) = \emptyset$, then

$$DIF = \langle p_{i_{k-2}} \rightarrow p_{i_{k-1}} \overset{l-1}{\rightarrow} p_{i_k}, \emptyset \rangle.$$

If $M(p_{i_{k-2}}, p_{i_k}, l-2) = \emptyset$ and $p_{i_{k-1}}$ conveys to p_{i_k} at $l-1$ that $M(p_{i_{k-2}}, p_{i_{k-1}}, l-2) = \emptyset$, then

$$DIF = \emptyset.$$

Here $M(p_{i_{k-2}}, p_{i_k}, l-2 | M(p_{i_{k-1}}, p_{i_k}, l-1)) = \emptyset$ is implicit in p_{i_k}'s transmission at l. Notice that if

$$M(p_{i_{k-2}}, p_{i_{k-1}}, l-2 | M(p_{i_{k-1}}, p_{i_k}, l-1)) = PIVOT(p_{i_{k-2}}, p_{i_k}, l-2),$$

then $DIF = \langle p_{i_{k-2}} \rightarrow p_{i_{k-1}} \overset{l-1}{\rightarrow} p_{i_k}, = \rangle$. The case where $p_{i_{k-1}}$ conveys to p_{i_k} at $l-1$ that it detected that $p_{i_{k-2}}$ was ill at $l-2$ is treated similarly.

Assume now that $p_{i_{k-1}}$ neither conveyed to p_{i_k} at $l-1$ that

$$M(p_{i_{k-2}}, p_{i_{k-1}}, l-2) = \emptyset$$

nor detected that $p_{i_{k-2}}$ was ill at $l-2$. Here, we construct DIF as follows. First, let

$$\begin{aligned} \alpha &= INPUT(p_{i_{k-2}}, l-3 | M(p_{i_{k-1}}, p_{i_k}, l-1)) \\ \alpha' &= INPUT(p_{i_{k-2}}, l-3 | PIVOT(p_{i_{k-2}}, p_{i_k}, l-2)). \end{aligned}$$

Then the atom

$$\langle p_{i_{k-2}} \rightarrow p_{i_{k-1}} \overset{l-1}{\rightarrow} p_{i_k}, \alpha \rangle$$

is included in DIF iff $\alpha \neq \alpha'$.

Second, consider the atom

$$a = \langle p_{i_1} \to \ldots \to p_{i_{k-2}} \xrightarrow{l-2} p_{i_{k-1}}, \alpha\rangle \in M(p_{i_{k-2}}, p_{i_{k-1}}, l-2|M(p_{i_{k-1}}, p_{i_k}, l-1)).$$

If for no $1 \leq f < k-2$ is the atom

$$a_f = \langle p_{i_f} \to \ldots \xrightarrow{l-3} p_{i_{k-2}}, \beta\rangle$$

conveyed in $PIVOT(p_{i_{k-2}}, p_{i_k}, l-2)$, where $\beta = \emptyset$ or $\beta =$"detected ill", then let

$$a' = \langle p_{i_1} \to \ldots \xrightarrow{l-3} p_{i_{k-2}}, \alpha'\rangle$$

be the corresponding atom conveyed in $PIVOT(p_{i_{k-2}}, p_{i_k}, l-2)$. In this case the atom

$$\langle p_{i_1} \to \ldots \to p_{i_{k-1}} \xrightarrow{l-1} p_{i_k}, \alpha\rangle$$

is included in DIF iff $\alpha \neq \alpha'$. If, on the other hand, a_f is conveyed in

$$PIVOT(p_{i_{k-2}}, p_{i_k}, l-2)$$

for some f, then a is included in DIF regardless of its content.

The case where

$$\langle p_{i_1} \to \ldots \xrightarrow{l-3} p_{i_{k-2}}, \alpha'\rangle \in PIVOT(p_{i_{k-2}}, p_{i_k}, l-2)$$

but $p_{i_{k-1}}$ conveys

$$\langle p_{i_1} \to \ldots \to p_{i_{k-2}} \xrightarrow{l-2} p_{i_{k-1}}, \alpha\rangle$$

to p_{i_k} at $l-1$, with $\alpha \neq \alpha'$, is treated similarly.

Having completed the operations above, we delete redundant information from DIF. For every atom

$$a = \langle p_{i_1} \to \ldots \xrightarrow{l-1} p_{i_k}, \alpha\rangle \in DIF$$

no atom of the form

$$\langle p_{i_1} \to s \to p_{i_2} \to \ldots \xrightarrow{l-1} p_{i_k}, \gamma\rangle$$

should be in DIF.

The operator $\bigtriangledown$ is roughly the inverse of $\bigtriangleup$. If M_3 denotes $M_1 \bigtriangleup M_2$, then $M_1 = M_3 \bigtriangledown M_2$.

The effect of $\bigtriangleup$ and $\bigtriangledown$ on transmission trees instead of on sets of atoms is now self-explanatory. Further, the time and space that these operators require when applied to transmission trees is *linear* in the sizes of the first and the second operands, up to multiplicative factors of polynomial size in n. The key idea is again a proper choice of the data structure. Every transmission tree will be represented as an n-ary tree, where each node may be connected to at most $n-1$ other nodes representing the other $n-1$ processors, and to a special node carrying the content corresponding to the chain defined by the path from the root to that node. We represent each such node by a vector of size n containing pointers to its sons. Locating a pointer to a son involves only $\mathcal{O}(\log n)$ time. Executing Δ involves visiting nodes in the transmission tree of the first operand in, e.g., depth first search order, and adding or deleting pointers in some nodes according to the second operand and the data structure described below.

A.5 The Data Structure

The data structure that each processor p that is healthy at a round l uses in every run ρ of NIP is called the *Necessary Knowledge Data Structure* and is denoted by $NK[\rho](p,l)$. It is an efficient data structure for encoding the knowledge of p at round l in ρ, and it is especially designed for allowing swift updates as new information flows in.

The data structure $NK[\rho](p,l)$ is a graph in which both the vertices and the edges are labelled. Each of its vertices denotes a pair (q,k), where q is a processor and k is a round, for $0 \leq k \leq l$. The vertex corresponding to (q,k) is labelled by the nc-state$[\rho](p,l)$ of q at round k, and is denoted $LB(q,k)$. The vertex corresponding to $(q,0)$ is labelled nc-healthy$[\rho](p,l)$ for all q.

There is an edge between two vertices (r,f) and (s,k) in the graph iff $r \neq s$ and $|f-k| = 1$. The label of the edge $((r,k-1),(s,k))$ is denoted by $LB(r,s,k)$.

If p knew at l that r was healthy at k, then $LB(r,p,l)$ would only carry $INPUT(r,k)$. Otherwise, $LB(r,p,k)$ carries only that part of the new information that r conveyed to p at k that p can trust at l only if p knew at l that r was healthy at k.

For $s \neq p$, $LB(r,s,k)$ is undefined unless p knows at l that s was healthy at $k+1$. In that case $LB(r,s,k)$ carries the new information that s conveyed to p at $k+1$ about $M(r,s,k)$, that is, the difference between $M(r,s,k)$ and $PIVOT(r,p,k)$.

A.6 The Procedure for Updating NK

Processor p inductively constructs $NK[\rho](p,l)$ from both $NK[\rho](p,l-1)$ and the messages it receives at round l as follows:

Base $(l=1)$: For every $(q,0)$, $LB(q,p,1) \leftarrow M(q,p,1)$. All the other edges in $NK[\rho](p,1)$ are not labelled.

Step: Assume inductively that $NK(p,l-1)$ was already built. Construct $NK(p,l)$ by invoking the routine UPDATE_NK(p,l) that is described below.

This routine performs three basic tasks: First, for every q so that $M(q,p,l) \neq \emptyset$, p trusts the information that q conveyed to p at $l-1$. Next, for every q so that $M(q,p,l) \neq \emptyset$, p examines the consistency of the message that q transmits to it at l. Finally, for every processor q so that either $(q,l-1) \in$ nc-pseudo-healthy(p,l) or $(q,l-1) \in$ nc-pseudo-dead(p,l), p checks if there are enough witnesses to prove that q was in fact ill at $l-1$.

```
Procedure UPDATE_NK$(p,l)$
;This procedure constructs $NK(p,l)$ based on $NK(p,l-1)$ and on
;the messages that $p$ receives at round $l$.
  For every $(q,k)$. $k < l$
      $LB(q,k)$ in $NK(p,l) \leftarrow LB(q,k)$ in $NK(p,l-1)$.

; For every $q$ so that $M(q,p,l) \neq \emptyset$. trust the information
; that $q$ conveyed at $l-1$.
  For every $q$ satisfying $M(q,p,l) \neq \emptyset$
      TRUST$(q,l-1)$

; For every $q$ so that $M(q,p,l) \neq \emptyset$. examine the consistency
; of the message $M(q,p,l)$.
  For every $q$ satisfying $M(q,p,l) \neq \emptyset$
      EXAMINE$(q,p,l)$

; For every $q$ so that either $(q,l-1) \in$ nc-pseudo-healthy$(p,l)$ or
; $(q,l-1) \in$ nc-pseudo-dead$(p,l)$. check if there are enough
; witnesses to prove that $q$ was ill at $l-1$.
  For every $q$ s.t. $LB(q,l-1) =$ nc-pseudo-healthy$(p,l)$
      If CHECK_ILL$(q,p,l)$ then $LB(q,l-1) \leftarrow$ nc-ill$(p,l)$
  For every $q$ s.t. $LB(q,l-1) =$ nc-pseudo-dead$(p,l-1)$
      If CHECK_ILL$(q,p,l)$ then
        $LB(q,l-1) \leftarrow$ nc-ill$(p,l)$.
        TRUST$(q,l-2)$.
  RETURN
```

Routine TRUST(q,k)

```
; This procedure adopts the information transmitted by $(q,k)$.
  If $LB(q,k) =$ nc-healthy$(p,l)$ then RETURN
  Else
    $LB(q,k) \leftarrow$ nc-healthy$(p,l)$
    For every son $r$ of $LB(q,p,k)$'s root
        $LB(r,q,k-1) \leftarrow ST(r,q,k-1|LB((q,p,k))$
        If $LB(r,q,k-1) \neq \emptyset$ then
          If $LB(r,k-1) =$ nc-pseudo-healthy$(p,l)$ then
            $LB(r,k-1) \leftarrow$ nc-ill$(p,l)$
          If $LB(r,k-1) =$ nc-pseudo-dead$(p,l)$ then
            $LB(r,k-1) \leftarrow$ nc-ill$(p,l)$
            TRUST$(r,k-2)$
    For every $r$ s.t. $LB(r,k+1) =$ nc-pseudo-healthy$(p,l)$
        If $r$ lied about $M(q,r,k)$ at $k+1$. i.e., $ST(q,r,k|LB(r,p,k+1)) \neq \emptyset$.
          then $LB(r,k+1) \leftarrow$ nc-ill$(p,l)$
    $LB(q,p,k) \leftarrow INPUT(q,k)$
    RETURN
```

```
Routine CHECK_ILL$(q,p,l)$
;This procedure checks whether there are sufficient witnesses
;for letting $LB(q,l-1)$ be nc-ill$(p,l)$ instead of either
;nc-pseudo-healthy$(p,l)$ or nc-pseudo-dead$(p,l)$.
  Let $FAIL = |$nc-failing$(p,l)|$.
  If there are *more* than $t - FAIL$ processors $r$. $r \neq p$. $r \neq q$.
  s.t. $M(q,r,l-1|M(r,p,l)) \neq M(q,r,l-1|M(q,p,l-1)$.
  i.e. $ST(q,r,l-1|LB(r,p,l)) \neq \emptyset$
      then RETURN(TRUE)
      else RETURN(FALSE)
```

```
Routine EXAMINE$(q,p,l)$
; Examine the consistency of $M(q,p,l)$ by showing that for every $r$
; at $l-1$, $q$ conveyed at $l$ consistent information about that $r$.
; Thereafter, set both $LB(q,l)$ and $LB(q,p,l)$.
  Check the syntax of the message
  $LB(q,p,l) \leftarrow M(q,p,l)$
  For every $r$. $r \neq q$. $r \neq p$
```

```
    CASE
      $LB(r,l-1)$= nc-healthy$(p,l)$
                  If $M(r,q,l-1|M(q,p,l)) = M(r,q,l-1|M(r,p,l-1))$
                    then $LB(q,p,l) \leftarrow PRUNE(r,q,l-1|LB(q,p,l))$
                  Else DETECT_ILL$(q,p,l)$; RETURN
      $LB(r,l-1)$= nc-dead$(p,l)$
                  If $M(r,q,l-1|M(q,p,l) = \emptyset$
                    then $LB(q,p,l) \leftarrow PRUNE(r,q,l-1|LB(q,p,l))$
                  Else DETECT_ILL$(q,p,l)$; RETURN
      Else,
                  If $\neg$NEW_INFORMATION$(r,q,p,l)$
                    then DETECT_ILL$(q,p,l)$; RETURN
                  $TREE \leftarrow ST(r,q,l-1|LB(q,p,l))$
                  If $\neg$CONSISTENT$(r,l-2,TREE)$
                    then DETECT_ILL$(q,p,l)$; RETURN
    ENDCASE
  SET_LABEL$(q,p,l)$
  $LB(q,l) \leftarrow$ nc-pseudo-healthy$(p,l)$
  RETURN
```

```
Routine NEW_INFORMATION$(r,q,p,l)$
; Check that $q$ conveyed to $p$ at $l$ only new information about $r$.
  For every atom $\langle \ldots \rightarrow s \rightarrow r \overset{l-1}{\rightarrow} q, \alpha \rangle \in M(q,p,l)$
    Let $\langle \ldots \rightarrow s \overset{l-2}{\rightarrow} q, \beta \rangle$ be the corresponding atom in $RV(q,l-2)$
    If $\alpha = \beta$ then RETURN(FALSE)
; Check that atoms carrying $\emptyset$ content were conveyed only when
; really needed.
  For every $\langle p_{i_1} \rightarrow \ldots \rightarrow p_{i_k} \overset{l-1}{\rightarrow} q, \emptyset \rangle \in M(q,p,l)$, where $p_{i_k} = r$
    If $\langle p_{i_1} \rightarrow s \rightarrow p_{i_3} \rightarrow \ldots \overset{l-1}{\rightarrow} q, \alpha \rangle \in M(q,p,l)$
      then RETURN(FALSE)
  RETURN(TRUE)
```

```
Routine CONSISTENT$(r,k,TREE)$
; Check the consistency of the information that $r$ claimed to have
; received at $k$, assuming that the information conveyed in $TREE$
; is reliable.  Notice that the root of $TREE$ is labelled $r$.
;
```

```
; Find the nc-known$(r,k|TREE)$ processors at $k-1$ based on
; $RV(r,k-1)$ and assuming the information in $TREE$.
  For every $s$ satisfying $M(s,r,k|TREE) \neq \emptyset$
    If$(s,k-1) \in$ nc-pseudo-healthy$(r,k-1)$
      then TRUST$(s,k-1)$
      else RETURN(FALSE)
; Check the consistency of the messages that $r$ received at $k$.
; First check that if $(w,k-1) \in$ nc-known$(r,k|TREE)$ then there
; is no chain in $TREE$ of the form $\ldots \rightarrow w \rightarrow s \xrightarrow{k} r$.
CHECK:
  For every $(w,k-1) \in$ nc-known$(r,k|TREE)$,
    If a son of $TREE$'s root has a son labelled $w$
      then RETURN(FALSE)
; Next check that processor $r$ checked at $k$ the consistency of
; the message that it received from every son $s$ of TREE's root.
  For every son $s$ of $TREE$'s root
    $NEW\_TREE \leftarrow ST(s,r,k|TREE)$
    If $\neg$CONSISTENT$(s,k-1,NEW\_TREE)$ then RETURN(FALSE)
  RETURN(TRUE)
```

```
Routine DETECT_ILL$(q,p,l)$
  $LB(q,l) \leftarrow$ " detected ill"
  $LB(q,p,l) \leftarrow$ " detected ill"
  RETURN
```

```
Routine SET_LABEL$(q,p,l)$
  Create $LB(q,p,l)$ by first constructing a single node tree, with root $\nu$ labeled $q$.
  Next, append a son to that node with label $INPUT(q,l|M(q,p,l))$.
  Finally, for every $r$, $r \neq p$ and $r \neq q$, s.t. $LB(r,l-1)$ is neither nc-healthy$(p,l)$
  nor nc-dead$(p,l)$, append the root of the tree
          $M(r,q,l-1|M(q,p,l)) \bigtriangleup PIVOT(r,p,l-1)$
  to $\nu$. This tree can be computed efficiently using the formula in section A.8.
  RETURN
```

A.7 The Transmission Procedure

The transmission procedure for processor p at round $l+1$ in a run of NIP involves two steps: First, p constructs the data structure $NK(p,l)$ by updating $NK(p,l-1)$ according to the messages that it received at round l. Next, for every s such that $LB(s,l)$ is either nc-pseudo-healthy(p,l) or nc-detected-ill(p,l), it merges the labels $\{LB(q,p,l)\}_{q\neq p,s}$ thereby creating the transmission tree $TT(p,s,l+1)$. If, on the other hand, p has seen t faulty processors by the end of round l, it knows that all the transmitting processors were healthy. Thus the only information it transmits at $l+1$ is its $INPUT$. The transmission procedure follows.

```
Procedure NIP_MESSAGES($p,l+1$)
;This procedure generates $p$'s transmission at $l+1$.
  UPDATE_NK($p,l$)
  For every $s$
      If $LB(s,l) =$ nc-pseudo-healthy($p,l$) then TRANSMIT($p,s,l+1$)
      If $LB(s,l) =$ nc-healthy($p,l$) then $TT(p,s,l+1) \leftarrow INPUT(p,l)$
  RETURN
```

```
Procedure TRANSMIT($p,s,l+1$)
;This procedure generates $M(p,s,l+1)$.
  Create $TT(p,s,l+1)$ by first constructing a single node tree, with root $\nu$
  labelled $p$. Next, append a son to that node with label $INPUT(p,l)$.
  Now, for every $r$, $r \neq p$ and $r \neq s$, make the root of $LB(r,p,l)$
  a son of $\nu$.
  RETURN
```

A.8 Routine SET_LABEL

In this section we indicate roughly how to evaluate the tree $LB(q,p,l)$ used in routine SET_LABEL. Notice first that

$$ST(s,r,l-2|LB(q,p,l)) \cong M(s,r,l-2|M(q,p,l)) \bigtriangleup M(s,r,l-2|M(r,p,l-1)).$$

We consider the two terms on the right hand side.

By definition,

$$M(s,r,l-2|M(q,p,l)) = ST(s,r,l-2|M(q,p,l) \triangledown PIVOT(s,q,l-2|M(q,p,l-1)).$$

Assume for the moment that

$$PIVOT(s,q,l-2|M(q,p,l-1)) = M(s,q,l-2).$$

The case where the two terms above are different is treated similarly.

Recall that

$$M(s,q,l-2) = LB(s,q,l-2) \bigtriangledown PIVOT(s,p,l-2),$$

and using the associativity of $\bigtriangledown$,

$$\begin{aligned}&M(s,r,l-2|M(q,p,l)) = \\ &\qquad (ST(s,r,l-2|M(q,p,l) \bigtriangledown LB(s,q,l-2)) \bigtriangledown PIVOT(s,p,l-2).\end{aligned}$$

Also,

$$M(s,r,l-2|M(r,p,l-1)) = ST(s,r,l-2|LB(r,p,l-1)) \bigtriangledown PIVOT(s,p,l-2).$$

But we have the following relation,

$$(M_1 \bigtriangledown M) \bigtriangleup (M_2 \bigtriangledown M) = M_1 \bigtriangleup M_2,$$

thus,

$$\begin{aligned}&ST(s,r,l-2|LB(q,p,l)) \cong \\ &\qquad \{ST(s,r,l-2|M(q,p,l)) \bigtriangledown LB(s,q,l-2)\} \bigtriangleup ST(s,r,l-2|LB(r,p,l-1)).\end{aligned}$$

The time and space used in evaluating this formula is estimated in the proof of lemma 13 in appendix E.2.

B The Consistency Test in NIP

In this section we prove that the consistency test of messages performed in NIP is as effective as the most general consistency test. More specifically, let p be healthy at l in a run of NIP. Suppose that p tries to determine whether q was ill at l by examining the message that q transmitted to it at l. It is fairly simple for p to check whether q forwarded correctly the messages that q received at $l-1$ from each r, so that p knows at l that r was either healthy or dead at $l-1$. When p does not know at l that r was either healthy or dead at $l-1$, then the only facts that p may and will check, are that q transmitted only new information about r and that q checked at $l-1$ the reliability of the message that r transmitted to it.

In lemma 2 we prove that p need only check that q checked the consistency of the new information that q conveyed to p at l about r. To this end consider an Extended New Information Protocol called ENIP. The routines that define ENIP are identical to the routines of NIP with one exception: In ENIP the routine CONSISTENT checks all the atoms that are conveyed instead of checking just the new information.

Here is the routine CONSISTENT used in ENIP:

```
Routine CONSISTENT$(r, k, TREE)$
; This is routine CONSISTENT in ENIP.
; Check the consistency of the information that $r$ claimed to have
; received at $k$, assuming that the information conveyed through
; $TREE$ is correct.
;
; Find the nc-known$(r, k|TREE)$ processors at $k-1$ based on
; $RV(r, k-1)$ and assuming the information in $TREE$.
  For every $s$ satisfying $M(s, r, k|TREE) \neq \emptyset$
    If$(s, k-1) \in$ nc-pseudo-healthy$(r, k-1)$
     then TRUST$(s, k-1)$
     else RETURN(FALSE)
; Check the consistency of the messages that $r$ received at $k$.
; First check that if $(w, k-1) \in$ nc-known$(r, k|TREE)$, then no
; atom of the form $\langle \ldots \rightarrow w \rightarrow s \xrightarrow{k} r, \alpha \rangle$ was conveyed by $s$ to
; $r$ through $M(s, r, k|TREE)$ that is inconsistent with
; $(w, k-1) \in$ nc-known$(r, k|TREE)$.
; In NIP the following test is performed instead:
; If a son of $TREE$'s root has a son labelled $w$
;    then RETURN(FALSE).
CHECK:
  For every $(w, k-1) \in$ nc-known$(r, k|TREE)$
    If some $s$ conveyed to $r$ at $k$ an atom of the form
    $\langle \ldots \rightarrow w \rightarrow s \xrightarrow{k} r, \alpha \rangle$ through $M(s, r, k|TREE)$
    that is inconsistent with $(w, k-1) \in$ nc-known$(r, k|TREE)$
      then RETURN(FALSE)
; Next check that processor $r$ checked at $k$ the consistency
; of all the messages that it received.
; In NIP the following ''for'' statement is performed instead:
; For every son $s$ of $TREE$'s root
  For every $s$
    $NEW\_TREE \leftarrow ST(s, r, k|TREE)$
    If $\neg$CONSISTENT$(s, k-1, NEW\_TREE)$ then RETURN(FALSE)
  RETURN(TRUE)
```

In lemma 2 we state that this strong consistency check is no more effective than the consistency check in NIP. In fact, p detects at l that q is ill at l in ENIP iff it does so in NIP.

Lemma 2 *For given INPUT, CA and appropriate ADV, let ρ and ρ^E be runs of NIP and ENIP with these parameters. Then $EX(\rho) = EX(\rho^E)$.*

Proof: For typographic reasons denote CONSISTENT and $TREE$ by CN and TR respectively.

We prove by induction on the round number l that all the processors transmit exactly the same messages in both ρ and ρ^E.

Base $l = 1$: Trivial since CN is never invoked.

Inductive step: Assume that the processors transmit exactly the same messages up to and including round l in both runs. We now prove that they will also transmit the same messages at $l + 1$.

This claim is trivial for ill processors at $l + 1$ as well as for dead processors there. We are therefore left with the healthy processors at $l + 1$. For every such healthy processor at $l + 1$ we must show that at l that processor had precisely the same reduced view in ρ and in ρ^E.

Let $p_{i_{k+1}}$ be such a healthy processor at $l + 1$, and assume that processor p_{i_k} transmits to $p_{i_{k+1}}$ at l. If p_{i_k} was either healthy or dead, there are no problems. The case that does require careful examination is when p_{i_k} is ill at l. In fact we must show that $(p_{i_k}, l) \in$ nc-detected-ill$[\rho](p_{i_{k+1}}, l)$ iff $(p_{i_k}, l) \in$ nc-detected-ill$[\rho^E](p_{i_{k+1}}, l)$.

The "if" case is trivial. We concentrate therefore on the "only if" part. We prove the following slightly involved claim by induction on j, the depth of the recursive invocation to CN.

1. For every depth j sequence of recursive invocations to CN with parameters

   $$(p_{i_{k-1}}, l-2, TR_{k-1}), (p_{i_{k-2}}, l-3, TR_{k-2}), \ldots, (p_{i_{k-j}}, l-j-1, TR_{k-j})$$

   in ρ there exists exactly the same sequence in ρ^E.

2. CN$(p_{i_{k-j}}, l-j-1, TR_{k-j})$ returns FALSE in ρ at step **CHECK** iff CN$(p_{i_{k-j}}, l-j-1, TR_{k-j})$ does in ρ^E.

3. If there exists a depth j sequence of recursive invocations to CN with parameters

   $$(p_{i_{k-1}}, l-2, TR_{k-1}), (p_{i_{k-2}}, l-3, TR_{k-2}), \ldots, (p_{i_{k-j}}, l-j-1, TR_{k-j})$$

   in ρ^E, and if there is no such sequence in ρ, then CN$(p_{i_{k-j}}, l-j-1, TR_{k-j})$ returns TRUE in ρ^E.

We now present the proof of the inductive hypothesis.

Base $j = 0$: NIP and ENIP are identical before invoking CN.

Inductive step: We proceed to prove 1, 2 and 3.

Proof of 1: By 1 in the inductive hypothesis, if there exists a depth $j - 1$ sequence of recursive invocations to CN in ρ then there exists exactly the same sequence in ρ^E. By 2 in the inductive hypothesis, the call to CN at depth $j - 1$ in that sequence in ρ returns FALSE at step **CHECK** iff it does so in ρ^E. By 3 in the inductive hypothesis, if there exists a recursive invocation to CN within the depth $j - 1$ call in that sequence in ρ^E that does not appear in ρ, then it returns TRUE in ρ^E. The statement now follows since whenever CN is recursively called in ρ, it is also called in ρ^E.

Proof of 2: This is the crux of the inductive claim. Obviously, if $(w, l-j-2) \in$ nc-known$(p_{i_{k-j}}, l-j-1)$ then CN returns FALSE at step **CHECK** in ρ when examining the atom

$$\langle \ldots \to w \to s \overset{l-j-1}{\to} p_{i_{k-j}}, \alpha \rangle$$

in TR_{k-j} iff it does so when examining the same chain in ρ^E.

Problems may arise therefore only while examining in ρ^E some atom

$$\langle \ldots \to w \to s \overset{l-j-1}{\to} p_{i_{k-j}}, \alpha \rangle$$

that does *not* appear in TR_{k-j} and so that $(w, l-j-2) \in$ nc-known$(p_{i_{k-j}}, l-j-1)$. In ρ^E this atom might create an inconsistency, whereas in ρ it is not checked at all. We now show that this atom need not be checked.

More formally, assume that in ρ the call to CN at depth j did not return FALSE at **CHECK**, whereas in ρ^E that same invocation returned FALSE. Thus, for some w such that $(w, l-j-2) \in$ nc-known$(p_{i_{k-j}}, l-j-1)$ and for some s so that the atom

$$\langle \ldots \to w \to s \overset{l-j-1}{\to} p_{i_{k-j}}, \alpha \rangle$$

is not in TR_{k-j}, this atom carries new information to $p_{i_{k-j}}$ at $l - j - 1$. Carrying new information means that the atom

$$\langle \ldots \to w \overset{l-j-2}{\to} p_{i_{k-j}}, \beta \rangle$$

satisfies $\beta \neq \alpha$.

We show this to be impossible by considering the following three cases:

Case 1: $M(w, p_{i_k}, l-j-1) \neq \emptyset$ (in both ρ and ρ^E).

Then $(w, l-j-2) \in$ nc-healthy$(p_{i_k}, l-j-1)$ and s conveyed correctly to p_{i_k} the information it received from w. Further, since the atom

$$\langle \ldots \to w \to s \overset{l-j-1}{\to} p_{i_{k-j}}, \alpha \rangle$$

is not in TR_{k-j}, s conveyed the same atom to $p_{i_{k-j}}$ and to p_{i_k} at $l-j-1$ (in $RV(p_{i_k}, l-1)$). By assumption the call to CN at depth j in ρ did not return FALSE at **CHECK**, thus $p_{i_{k-j}}$ conveyed

$$\langle \ldots \rightarrow w \rightarrow s \overset{l-j-1}{\rightarrow} p_{i_{k-j}}, \alpha \rangle$$

at $l-j-1$, which is *not* new information – a contradiction.

Case 2: $M(w, p_{i_k}, l-j-1) = \emptyset$ and $M(w, p_{i_{k-j}}, l-j-1) \neq \emptyset$.

Since $M(w, p_{i_{k-j}}, l-j-1) \neq \emptyset$ and since the call to CN at depth j in ρ did not return FALSE at **CHECK**, then atoms of the form

$$\langle \ldots \rightarrow w \rightarrow s \overset{l-j-1}{\rightarrow} p_{i_{k-j}}, \alpha \rangle$$

cannot carry new information to $p_{i_{k-j}}$ – a contradiction.

Case 3: $M(w, p_{i_k}, l-j-1) = \emptyset$ and $M(w, p_{i_{k-j}}, l-j-1) = \emptyset$.

$M(w, p_{i_{k-j}}, l-j-1) = \emptyset$ implies that $(w, l-j-2) \in$ nc-dead$(p_{i_{k-j}}, l-j-1)$, and the meaning of the atom that makes CN return FALSE at **CHECK** in ρ^E is that s conveyed to $p_{i_{k-j}}$ (and also to p_{i_k}) at $l-k-1$ that it received some nonempty message from w. We proceed to show that this leads to a contradiction.

If for some $f \leq l-j-2$, $(w, f) \in$ nc-dead$(p_{i_{k-j}}, l-j-3)$, then $p_{i_{k-j}}$ conveyed that fact to p_{i_k} at $l-j-2$, and therefore p_{i_k} believed it at $l-j-1$. Thus, p_{i_k} also knew at $l-j-1$ that w was dead at $l-j-2$. Now s conveyed that same atom also to p_{i_k}, therefore the healthy p_{i_k} at $l-j-1$ should also have discovered that s was ill at $l-j-1$, which it did not – a contradiction.

Otherwise, there is no such f, and in particular, $p_{i_{k-j}}$ did not know at $l-j-3$ that w would be dead at $l-j-2$. Assume next that $p_{i_{k-j}}$ discovered at $l-j-2$ that w was dead at $l-j-2$, that is, $(w, l-j-2) \in$ nc-dead$(p_{i_{k-j}}, l-j-2)$. Thus $p_{i_{k-j}}$ discovered only at $l-j-2$ that w was ill at $l-j-3$. This could have happened only after invoking the procedure CHECK_ILL within procedure UPDATE_NK. It follows that there are more than $t - FAIL$ messages that are different from $M(w, p_{i_{k-j}}, l-j-3)$ that were transmitted to $p_{i_{k-j}}$ at $l-j-2$. At least one of the processors that transmitted such a message, say v, was healthy at $l-j-2$ and also at $l-j-1$. Thus p_{i_k} knew at $l-j-1$ at least two different versions of the messages that w transmitted at $l-j-3$: One from $p_{i_{k-j}}$ at $l-j-2$ and the other from v at $l-j-2$.

Recall now that p_{i_k} knew at $l-j-1$ that both $p_{i_{k-j}}$ and v were healthy at $l-j-2$, therefore it also must have known that w was ill at $l-j-3$ and therefore dead at $l-j-2$. Thus p_{i_k} should have detected at $l-j-1$ that s was ill at $l-j-1$ – a contradiction.

We are left with one more case: $(w, l-j-2) \notin$ nc-dead$(p_{i_{k-j}}, l-j-2)$. Thus $p_{i_{k-j}}$ discovered only at $l-j-1$ that w was ill at $l-j-3$. Another look at routine UPDATE_NK reveals that there must have been at least one call to procedure TRUST, and therefore there must have been some v so that $M(v, p_{i_{k-j}}, l-j-1) \neq \emptyset$. Thus $p_{i_{k-j}}$ trusted $M(v, p_{i_{k-j}}, l-j-2)$ and inferred thereby that w was ill at $l-j-3$.

Recall that by assumption the call to CN at depth j in ρ did not return FALSE at **CHECK**, thus the atom

$$\langle \ldots \rightarrow w \rightarrow s \overset{l-j-1}{\longrightarrow} p_{i_{k-j}}, \alpha \rangle$$

cannot carry new information to $p_{i_{k-j}}$ – a contradiction.

This completes the proof of item 2 of the inductive hypothesis.

Proof of 3: Assume there was a depth j sequence of recursive invocations to CN with parameters

$$(p_{i_{k-1}}, l-2, TR_{k-1}), (p_{i_{k-2}}, l-3, TR_{k-2}), \ldots, (p_{i_{k-j}}, l-j-1, TR_{k-j})$$

in ρ^E, and that this sequence is absent from ρ. Thus there exists a minimal f, $0 < f \leq j$ so that CN$(p_{i_{k-f}}, l-f-1, TR_{k-f})$ is invoked in ρ^E, but not in ρ. Therefore CN$(p_{i_{k-f+1}}, l-f, TR_{k-f+1})$ (or EXAMINE$(p_{i_k}, p_{i_{k+1}}, l)$ if $f=0$) is called in both runs. Since there was no recursive call to CN$(p_{i_{k-f}}, l-f-1, TR_{k-f})$ in ρ, the subtree that $p_{i_{k-f+1}}$ transmitted about $p_{i_{k-f}}$ (in $RV(p_{i_k}, l-1)$) was either empty, "detected ill" or a single node labelled $p_{i_{k-f}}$ carrying an $INPUT$. The interesting case is the third. If $f=1$, then $p_{i_{k-1}}$ and p_{i_k} receive the same messages at $l-2$ in $RV[\rho^E](p_{i_k}, l-1)$, up to the weak information symmetry of NIP. Thus, the recursive call to CN$(p_{i_{k-f}}, l-f-1, TR_{k-f})$ in ρ^E returns true iff p_{i_k} checks the consistency of the messages that it receives at $l-2$ in ρ^E. But since $M[\rho^E](p_{i_k}, p_{i_{k-1}}, l) \neq \emptyset$, p_{i_k} is healthy at $l-1$ in ρ^E, and therefore it certainly checked the consistency of the messages that it received at $l-2$. The argument for $f > 1$ is similar. Just notice that $p_{i_{k-f}}$ and p_{i_k} receive the same messages at $k-f-1$ in $RV[\rho^E](p_{i_k}, l-1)$, up to differences due to weak information symmetry. This completes the proof of item 3 of the inductive hypothesis. ▮

C CK Characterization in NIP

In this section we introduce the critical round of a run ρ at a round l which we denote $CR[\rho](l)$. It plays a central role in the classification of the facts that are common knowledge at round l in the run ρ of NIP or of other protocols. Refer to [DM] and [MT] for a similar definition in the crash and the omission models respectively.

Let $N[\rho](k)$ be the number of processors that fail at k in ρ. Let the *segment critical round of run ρ at round l*, sg-$CR[\rho](l)$, be the smallest round number j such that the following threshold inequalities are satisfied:

$$t - N[\rho](k) \geq l - k \quad \text{for} \quad j \leq k \leq l.$$

The *critical round of ρ at l*, $CR[\rho](l)$, is defined by :

$$CR[\rho](l) = \min_{\rho' \overset{l}{\sim} \rho} \text{sg-}CR[\rho'](l).$$

Roughly speaking, the basic property of $CR[\rho](l)$ is that the states, $INPUT$s and transmissions of each processor q at k, so that $k > CR[\rho](l)$ and q does not fail at $CR[\rho](l)$, are ***not*** common knowledge at l in ρ. Refer to [MT] for more details on the relation between facts that are common knowledge at a round and the critical round corresponding to that round.

C.1 CK Evaluation in NIP

In this section we develop a procedure that allows every processor that is healthy at a round in a run of NIP to evaluate the critical round.

The definition of the critical round indicates that evaluating $CR[\rho](l)$ might involve checking all the runs in the l-similar equivalence class of ρ. Surprisingly, each processor p that is healthy at a round l in ρ need only consider runs that are (p,l)-equivalent to ρ for performing that evaluation. Fortunately, the data structure $NK[\rho](p,l)$ naturally engenders a method for calculating the critical round at l.

The idea behind the procedure that p uses in order to evaluate the critical round is as follows: After having assigned nc-states(p,l) to every pair (processor, round), p assigns at l another type of state which we call the *pr-state*(p,l). There are basically three different types of pr-state(p,l): pr-healthy(p,l), pr-ill(p,l) or pr-dead(p,l). The crux of the problem is the choice of pr-state(p,l) for pairs that are either nc-pseudo-healthy(p,l) or nc-pseudo-dead(p,l).

C.2 A Procedure for Evaluating the CR

The procedure that processor p uses at round l for evaluating the critical round at l follows:

```
Procedure CR($l$)
  For every $(q,k)$
    $(q,k)$'s pr-state$(p,l)$ $\leftarrow$ $(q,k)$'s nc-state$(p,l)$
  EVAL_CR($l$)
end

Procedure EVAL_CR($k$)
  $k \leftarrow$ JUMP($k$)
  If there exists some $m$ so that $(p_m,k) \in$ pr-pseudo-dead$(p,l)$
    then $LB(p_m,k) \leftarrow$ pr-ill$(p,l)$
    TRUST$(p_m, k-1)$
    EVAL_CR$(k-1)$
  Else RETURN("$CR(l) = k$")

Procedure JUMP($k$)
  Let $\delta \leftarrow t - |\{q \mid (q,k) \in$ pr-dead$(p,l) \cup$ pr-pseudo-dead$(p,l)\}|$
  If $\delta > l - k$ then RETURN((JUMP$(l-\delta)$)
  Else RETURN($k$)
```

This procedure for evaluating common knowledge readily generalizes to deterministic protocols other than NIP.

D The Proof of Theorem 1

We begin this appendix by proving the following justification of our definition of an information symmetric protocol:

Lemma 3 *The protocol $\mathcal{F} = \{F_{(p,l)}\}$ is information symmetric iff for every pair of runs σ and σ' of $\mathcal{F}$ in which p is healthy at l, for all processors q and r, $M[\sigma](p,q,l) = M[\sigma'](p,q,l)$ iff $M[\sigma](p,r,l) = M[\sigma'](p,r,l)$.*

Proof: $\Rightarrow$ Assume that $M[\sigma](p,q,l) = M[\sigma'](p,q,l)$. Then

$$F^q_{(p,l)}(V[\sigma](p,l-1)) = F^q_{(p,l)}(V[\sigma'](p,l-1))$$

implying that

$$(F^q_{(p,l)})^{-1} \circ F^q_{(p,l)}(V[\sigma](p,l-1)) = (F^q_{(p,l)})^{-1} \circ F^q_{(p,l)}(V[\sigma'](p,l-1)).$$

Since $\mathcal{F}$ is information symmetric,

$$(F^r_{(p,l)})^{-1} \circ F^r_{(p,l)}(V[\sigma](p,l-1)) = (F^r_{(p,l)})^{-1} \circ F^r_{(p,l)}(V[\sigma'](p,l-1)).$$

Now, this implies that

$$F^r_{(p,l)}(V[\sigma](p,l-1)) = F^r_{(p,l)}(V[\sigma'](p,l-1))$$

and therefore $M[\sigma](p,r,l) = M[\sigma'](p,r,l)$.

$\Leftarrow$ Let $V[\sigma](p,l-1)$ be a view; thus, by definition, p is healthy at l in σ. Let

$$V[\sigma'](p,l-1) \in (F^q_{(p,l)})^{-1} \circ F^q_{(p,l)}(V[\sigma](p,l-1))$$

where again p is healthy at l in σ'. Then

$$F^q_{(p,l)}(V[\sigma](p,l-1)) = F^q_{(p,l)}(V[\sigma'](p,l-1))$$

implying that $M[\sigma](p,q,l) = M[\sigma'](p,q,l)$. Applying the assumption, we have $M[\sigma](p,r,l) = M[\sigma'](p,r,l)$, which implies in turn that

$$V[\sigma'](p,l-1) \in (F^r_{(p,l)})^{-1} \circ F^r_{(p,l)}(V[\sigma](p,l-1)). \qquad \blacksquare$$

The following lemma redefines our notion of conveying by replacing each knowledge operator with an universal quantifier.

Lemma 4 *Assume that p is healthy at l in ρ and $\rho \models K_{(p,l-1)}\varphi$.*

p conveys φ to q at l in ρ iff, for every $\rho' \overset{(p,l-1)}{\approx} \rho$ and for every $\rho'' \overset{(q,l)}{\approx} \rho'$, so that p is healthy at l in ρ'', $\rho'' \models \varphi$.

We proceed now to prove theorem 1 by the following lemmas:

Lemma 5 *If a protocol $\mathcal{F}$ is an $\mathcal{RCP}$, then $\mathcal{F}$ is ck-informative.*

Proof: Following the notations of definition 2, let $\mathcal{F} = \{F_{(p,l)}\}$ be an $\mathcal{RCP}$. Let ρ be a run of $\mathcal{F}$, let ρ' satisfy $\rho' \overset{w\text{-}(p,l)}{\approx} \rho$, and let φ be a basic predicate such that $\rho' \models C_l\varphi$. We show that $\rho'' \models C_l\varphi$, where ρ'' differs from ρ' only in that p transmits at l in ρ'' using $F_{(p,l)}$.

Let $\mathcal{F}'$ and $\mathcal{F}''$ be the protocols in ρ' and ρ'' respectively. Pick an arbitrary run σ of $\mathcal{F}''$ satisfying $\sigma \overset{l}{\sim} \rho''$. Thus, for some runs σ_j of $\mathcal{F}''$, $j = 0, 1, \ldots, m$,

$$\sigma = \sigma_m \overset{(p_{i_m},l)}{\approx} \sigma_{m-1} \overset{(p_{i_{m-1}},l)}{\approx} \ldots \overset{(p_{i_2},l)}{\approx} \sigma_1 \overset{(p_{i_1},l)}{\approx} \sigma_0 = \rho''.$$

We may assume without loss of generality that $p_{i_j} \neq p_{i_{j+1}}$, for each $j = 1, \ldots, m-1$.

We successively modify each run σ_j into another run θ_j, also of $\mathcal{F}''$, so that the following conditions are satisfied for $j = 0, 1, \ldots, m$:

1. CA and $INPUT$ in θ_j and σ_j coincide.
2. All the messages in $SEG[\theta_j](l)$ are identical to the corresponding messages in $SEG[\sigma_j](l)$, excluding possibly messages $M(q,p,k)$, where $k < l$, such that $(q,k) \in$ nc-detected-ill$[\sigma_j](p,l)$.
3. If p is healthy at l in both θ_{j-1} and θ_j, then p has the *same view* at $l-1$ in both, i.e., $V[\theta_{j-1}](p,l-1) = V[\theta_j](p,l-1)$.
4. $\theta_{j-1} \overset{(p_{i_j},l)}{\approx} \theta_j$.

The salient point of this construction is item 3. Indeed, once the θ's are constructed, we will modify the protocol function of p at round l in these runs. Since whenever p is healthy at l in both θ_{j-1} and θ_j,

$$V[\theta_{j-1}](p,l-1) = V[\theta_j](p,l-1),$$

p will transmit precisely the same messages at l in the two resulting runs, thus maintaining the $\overset{(p_{i_j},l)}{\approx}$ relations. Here is the inductive construction of the θ's.

Base $j = 0$: Let $\theta_0 = \sigma_0$.

Inductive step: Assume that for $k = 0, 1, \ldots, j-1$, runs θ_k satisfying the conditions above were already constructed. We proceed to construct θ_j. Recall that $\sigma_{j-1} \overset{(p_{i_j},l)}{\approx} \sigma_j$, and consider the following two cases:

Case: $p_{i_j} \neq p$. Let q stand for p_{i_j}. Thus $V[\sigma_{j-1}](q,l) = V[\sigma_j](q,l)$. By item 2 of the inductive hypothesis, $V[\theta_{j-1}](q,l) = V[\sigma_{j-1}](q,l)$. Thus, $V[\theta_{j-1}](q,l) = V[\sigma_j](q,l)$.

Consider first the case where p is healthy at l in both θ_{j-1} and σ_j. Since $\mathcal{F}$ is an $\mathcal{RCP}$, $RV[\theta_{j-1}](p,l-1) = RV[\sigma_j](p,l-1)$. Now construct θ_j so that all the messages in $SEG[\theta_j](l)$ are identical to the corresponding messages in $SEG[\sigma_j](l)$, excluding possibly messages $M[\theta_j](q,p,k)$, for $k < l$, so that $(q,k) \in$ nc-detected-ill$[\sigma_j](p,l)$. Let each such message $M[\theta_j](q,p,k)$ be $M[\theta_{j-1}](q,p,k)$. It is apparent that this construction satisfies 1, 2 and 3.

Next we show that θ_j is a legitimate run of $\mathcal{F}''$. We argue first that p transmits precisely the same messages up to and including round l to corresponding processors in θ_{j-1} and σ_j. Indeed, this follows from $V[\theta_{j-1}](q,l) = V[\sigma_j](q,l)$ and from the fact that p conveys its reduced view at each round, and therefore it conveys all the messages it sends to all the other processors at that round. Second, by the construction of θ_j, p transmits, up to and including round l, precisely the same messages in both θ_{j-1} and θ_j. Thus, up to and including l, p transmits the same messages in both θ_j and σ_j. It follows that each processor other than p receives precisely the same messages in these two runs, and therefore it also transmits the same messages. This proves that θ_j is a legitimate run of $\mathcal{F}''$.

Finally, to show 4, recall that

$$\begin{aligned} V[\theta_{j-1}](q,l) &= V[\sigma_j](q,l) \\ V[\theta_j](q,l) &= V[\sigma_j](q,l). \end{aligned}$$

Thus, $\theta_{j-1} \overset{(q,l)}{\approx} \theta_j$.

Up to this point we have assumed that p is healthy at l in both θ_{j-1} and σ_j. Consider now the case where p is *not* healthy at l in one or both runs. In this case just let $\theta_j = \sigma_j$. Here conditions 1 to 3 are easily verified. Condition 4 is proved as follows: By 2 of the inductive hypothesis, $\theta_{j-1} \overset{(q,l)}{\approx} \sigma_{j-1}$. By definition, $\sigma_{j-1} \overset{(q,l)}{\approx} \sigma_j$. Thus, $\theta_{j-1} \overset{(q,l)}{\approx} \sigma_j$, and from the way we constructed θ_j, $\theta_{j-1} \overset{(q,l)}{\approx} \theta_j$.

Case: $p_{i_j} = p$. The treatment here is very similar to the one above. In fact, construct θ_j exactly as in the previous case. We now show that this construction is legitimate and that it satisfies 1 to 4.

To show that θ_j is a run of $\mathcal{F}''$, we need only state that p transmits in θ_j precisely as it does in σ_j. This is apparent from 2 of the inductive hypothesis, which implies that p transmits identically in θ_{j-1} and σ_{j-1}, and from $\sigma_{j-1} \overset{(p,l)}{\approx} \sigma_j$.

We proceed to prove 1 to 4. 1 and 2 hold from the way we constructed θ_j. By 2 of the inductive hypothesis, all the messages in $SEG[\theta_{j-1}](l)$ are identical to the corresponding messages in $SEG[\sigma_{j-1}](l)$, excluding possibly messages $M[\theta_j](q,p,k)$, for $k < l$, so that $(q,k) \in$ nc-detected-ill$[\sigma_{j-1}](p,l)$. From the way we constructed θ_j, for every such q and k, $M[\theta_j](q,p,k) = M[\theta_{j-1}](q,p,k)$, and all the other messages in $SEG[\theta_j](l)$ and $SEG[\sigma_j](l)$ are identical. But by assumption, $\sigma_{j-1} \overset{(p,l)}{\approx} \sigma_j$, that is $V[\sigma_{j-1}](p,l) = V[\sigma_j](p,l)$, thus we also have $V[\theta_{j-1}](p,l) = V[\theta_j](p,l)$. This proves 3 and 4.

For each $j = 0, 1, \ldots, m$, we successively construct a run λ_j of $\mathcal{F}'$ as follows:

- $INPUT$ and CA in θ_j and λ_j coincide.
- All the messages in $SEG[\lambda_j](l)$ are identical to the corresponding messages in $SEG[\theta_j](l)$, excluding possibly the messages that p transmits at l.

- If p is healthy at l in θ_j, then it transmits according to $\mathcal{F}'$ in λ_j.
- If p is ill at l in θ_j (and therefore $p \neq p_{i_j}$ and $p \neq p_{i_{j+1}}$) then:
  - If p is healthy at l in θ_{j-1}, then let
    $$M[\lambda_j](p, p_{i_j}, l) = M[\lambda_{j-1}](p, p_{i_j}, l).$$
  - If p is healthy at l in θ_{j+1}, then let
    $$M[\lambda_j](p, p_{i_{j+1}}, l) = M[\lambda_{j+1}](p, p_{i_{j+1}}, l).$$
  - If p is ill or dead at l in θ_{j+1}, then let
    $$M[\lambda_j](p, p_{i_{j+1}}, l) = M[\lambda_{j+1}](p, p_{i_{j+1}}, l) = \emptyset.$$

  Notice that these assignments of messages are always possible since, by assumption, $p_{i_j} \neq p_{i_{j+1}}$.

By item 3 in the construction of the θ's, and by the special treatment in the case that p is ill at l in some of the λ_j's, we conclude,

$$\lambda_m \overset{(p_{i_m}, l)}{\approx} \lambda_{m-1} \overset{(p_{i_{m-1}}, l)}{\approx} \ldots \overset{(p_{i_2}, l)}{\approx} \lambda_1 \overset{(p_{i_1}, l)}{\approx} \lambda_0 = \rho'.$$

But $\rho' \models C_l\varphi$, thus $\lambda_m \models \varphi$. Recall that for each $j = 0, 1, \ldots, m$, the CA and $INPUT$ in σ_j, θ_j and λ_j are identical. Thus, since φ is a basic predicate and $\sigma = \sigma_m$, $\sigma \models \varphi$. Therefore $\rho'' \models C_l\varphi$. ▮

Lemma 6 *If an information symmetric protocol $\mathcal{F} = \{F_{(p,l)}\}$ is ck-informative, then $\mathcal{F}$ is an $\mathcal{RCP}$.*

Proof: We show by induction on the round number l that the information symmetric and ck-informative protocol $\mathcal{F}$ is an $\mathcal{RCP}$.

Base (l=1): By definition, $RV(p, 0) = INPUT(p, 0)$ for every $p \in P$. The initial assumption about $\mathcal{F}$ implies that every processor that is healthy at 1 must convey its $INPUT$ and therefore also its reduced view.

Step: Assume inductively that every processor s that is healthy at k in σ conveys its reduced view for $k = 1, \ldots, l-1$, and assume by contradiction that p does not convey its RV to q at l in σ.

By lemma 4 there exist two runs σ' and σ'' of $\mathcal{F}$ so that $\sigma' \overset{(p,l-1)}{\approx} \sigma$, $\sigma'' \overset{(q,l)}{\approx} \sigma'$, p is healthy at l in σ'', and $RV[\sigma''](p,l-1) \neq RV[\sigma](p,l-1)$. But $V[\sigma'](p,l-1) = V[\sigma](p,l-1)$, implying that $RV[\sigma''](p,l-1) \neq RV[\sigma'](p,l-1)$.

The relations $\sigma' \overset{(p,l-1)}{\approx} \sigma$ and $\sigma'' \overset{(q,l)}{\approx} \sigma'$ imply $\sigma' \overset{(p,l-2)}{\approx} \sigma$ and $\sigma'' \overset{(q,l-1)}{\approx} \sigma'$ respectively. Applying the inductive hypothesis and since $l > 1$, p conveys its RV to all the other processors at $l-1$ in σ. Again by lemma 4, $RV[\sigma''](p,l-2) = RV[\sigma](p,l-2)$, implying $RV[\sigma''](p,l-2) = RV[\sigma'](p,l-2)$.

How could it happen that

$$RV[\sigma''](p,l-1) \neq RV[\sigma'](p,l-1)$$

but

$$RV[\sigma''](p,l-2) = RV[\sigma'](p,l-2)?$$

At least one of the following two situations must have occurred:

S1 There exists some processor r satisfying the following two conditions:

R1 $M[\sigma''](r,p,l-1) \neq M[\sigma'](r,p,l-1)$.

R2 It is not the case that both $(r,l-1) \in$ nc-detected-ill$[\sigma'](p,l-1)$ and $(r,l-1) \in$ nc-detected-ill$[\sigma''](p,l-1)$.

S2 $INPUT[\sigma''](p,l-1) \neq INPUT[\sigma'](p,l-1)$.

We argue that if either S1 or S2 holds, then $\mathcal{F}$ is not ck-informative.

Consider first the case where S1 holds, but S2 does not.

Let $\{r_i\}_{i=1}^{m}$ be the set of processors such that

$$M[\sigma''](r_i,p,l-1) \neq M[\sigma'](r_i,p,l-1).$$

Assume by contradiction that for some such r_j, $M[\sigma'](r_j,q,l) \neq \phi$. First, q knows at l in σ' that r_j was healthy at $l-1$. Second, $\sigma'' \overset{(q,l)}{\approx} \sigma'$, implying that $M[\sigma''](r_j,q,l-1) = M[\sigma'](r_j,q,l-1)$. Finally, from the information symmetry of $\mathcal{F}$ and by lemma 1, $M[\sigma''](r_j,p,l-1) = M[\sigma'](r_j,p,l-1)$, a contradiction.

Thus, $M[\sigma''](r_i,q,l) = M[\sigma'](r_i,q,l) = \phi$, for all $i = 1,\ldots,m$, implying that each $(r_i,l-1) \notin$ nc-healthy(q,l) in both σ'' and σ'.

R1 and R2 imply that one of the following two events must have occurred:

E1 For at least one $j \in \{1,\ldots,m\}$, $(r_j,l-1) \in$ nc-pseudo-healthy$(p,l-1)$ in either σ'' or σ'. Denote by r the r_j with smallest j satisfying the above.

E2 For all $i = 1, \ldots, m$, $M[\sigma'](r_i, p, l-1) = \phi$ and $(r_i, l-1) \in$ nc-detected-ill$[\sigma''](p, l-1)$ or vice versa (exchange σ'' and σ').

Consider E1 first. Assume without loss of generality that $(r, l-1) \in$ nc-pseudo-healthy$[\sigma'](p, l-1)$. Thus, there exists a run $\tilde{\sigma} \overset{(p,l-1)}{\approx} \sigma'$ in which r is healthy at $l-1$.

Construct a run ρ of $\mathcal{F}$ so that the messages in $SEG[\rho](l-1)$ coincide with the messages in $SEG[\tilde{\sigma}](l-1)$ excluding:

1. For $i = 1, \ldots, m$, let $M[\rho](r_i, p, l-1) = M[\sigma''](r_i, p, l-1)$. It follows therefore that $V[\rho](p, l-1) = V[\sigma''](p, l \cancel{+} 1)$.
2. Some t processors fail at $l-1$ in ρ. All the processors that fail at $l-1$ in ρ, excluding r, do not transmit to q at $l-1$.

Notice that the CA and $INPUT$ in ρ can be readily defined to satisfy the above, and that since t processors fail before l in ρ (including r_i, for $i = 1, \ldots, m$), ρ is completely specified.

We now prove that $\mathcal{F}$ is not ck-informative in ρ. Let ρ' be a run that differs from ρ only in that the processors that are healthy at l in ρ' transmit their corresponding views (at $l-1$). Note that ρ' has been designed so that $\rho' \overset{w\text{-}(p,l)}{\approx} \rho$. The processors that are healthy at l in ρ' receive t empty messages, thus, they know that the transmitting processors at l are healthy.

By the information symmetry of $\mathcal{F}$, each processor s that is healthy at l in ρ' knows that r was ill at $l-1$. The intuition is that since $M[\sigma''](r, p, l-1) \neq M[\sigma'](r, p, l-1)$, r manifests its illness at $l-1$ in ρ by transmitting messages that are inconsistent with the information symmetry of its protocol. To be more precise, note that if s assumes at l that r was healthy at $l-1$, then it may apply lemma 1 for calculating $M[\rho'](r, p, l-1)$ from $M[\rho'](r, s, l-1)$. Now, $M[\rho'](r, s, l-1) = M[\tilde{\sigma}](r, s, l-1)$, and $M[\rho'](r, p, l-1) = M[\sigma''](r, p, l-1)$. But $M[\sigma''](r, p, l-1) \neq M[\tilde{\sigma}](r, p, l-1)$; thus, since s knows at l that p was healthy at l it also knows that r must have been ill at $l-1$.

Recall that q conveys to all the processors that are healthy at l in ρ' that it received $t-1$ empty messages, hence each processor that is healthy at l in ρ' knows that t other processors failed at $l-1$. Thus, the views of the processors that are healthy at l in ρ' are common knowledge at l in ρ'. In particular, the basic predicate $\varphi \overset{\text{def}}{=}$ "r was ill at $l-1$" satisfies $\rho' \models C_l \varphi$.

Consider now ρ'' which differs from ρ' only in that p transmits at l according to $\mathcal{F}$ in ρ'' rather that transmitting its view as in ρ'. By construction,

$$V[\rho''](p, l-1) = V[\sigma''](p, l-1),$$

implying that $M[\rho''](p,q,l) = M[\sigma''](p,q,l)$. Recall that $V[\sigma''](q,l) = V[\sigma'](q,l)$ and $V[\sigma'](p, l-1) = V[\tilde{\sigma}](p, l-1)$, hence,

$$M[\sigma''](p,q,l) = M[\sigma'](p,q,l) = M[\tilde{\sigma}](p,q,l).$$

Thus, q does not know at l in ρ'' whether $M[\rho''](r,p,l-1)$ is $M[\tilde{\sigma}](r,p,l-1)$, in which case r could have been healthy at $l-1$ in ρ'', or $M[\rho''](r,p,l-1)$ is $M[\sigma''](r,p,l-1)$, in which case r was ill at $l-1$ in ρ''. Therefore, $\rho'' \models \neg C_l \varphi$, implying that $\mathcal{F}$ is not ck-informative.

Consider now E2: Assume without loss of generality that $M[\sigma'](r_1, p, l-1) = \phi$ and that $(r_1, l-1) \in$ nc-detected-ill$[\sigma''](p, l-1)$. Construct a run ρ, similar to the one above, so that the messages in $SEG[\rho](l-1)$ coincide with the messages in $SEG[\sigma'](l-1)$ excluding:

1. For $i = 1, \ldots, m$, let $M[\rho](r_i, p, l-1) = M[\sigma''](r_i, p, l-1)$. It follows therefore that $V[\rho](p, l-1) = V[\sigma''](p, l-1)$.
2. For $i = 1, \ldots, m$, r_i does not transmit at $l-1$ in ρ to any processor excluding p.
3. Some t processors do not transmit to q at $l-1$ in ρ.

Notice again that the CA and $INPUT$ in ρ can be readily defined to satisfy the above, and that since t processors fail before l in ρ (including r_i, for $i = 1, \ldots, m$), ρ is completely specified.

Let ρ' be a run that differs from ρ only in that the healthy processors at l in ρ' transmit their corresponding views. Note again that ρ' has been designed so that $\rho' \overset{w\text{-}(p,l)}{\approx} \rho$.

The healthy processors at l in ρ receive t empty messages, thus, they know that the transmitting processors at l are healthy. They know that r_1 was ill at $l-1$ from the transmissions of p at l and they know that t processors failed at $l-1$. Thus, the views of the healthy processors at l in ρ' are common knowledge at l in ρ'. In particular, the basic predicate $\varphi \overset{\text{def}}{=}$ "r_1 was ill at $l-1$" satisfies $\rho' \models C_l \varphi$.

Consider now ρ'' which differs from ρ' only in that p transmits at l according to $\mathcal{F}$ in ρ'' rather that transmitting its view as in ρ'. By construction

$$V[\rho''](p, l-1) = V[\sigma''](p, l-1),$$

implying that $M[\rho''](p,q,l) = M[\sigma''](p,q,l)$. Recall that $V[\sigma''](q,l) = V[\sigma'](q,l)$, hence, $M[\sigma''](p,q,l) = M[\sigma'](p,q,l)$. Thus, q does not know at l in ρ'' whether $M[\rho''](r_1,p,l-1)$ is $M[\sigma'](r_1,p,l-1)$, in which case r_1 could have been dead at $l-1$ in ρ'', or $M[\rho''](r_1,p,l-1)$ is $M[\sigma''](r_1,p,l-1)$, in which case r_1 was ill at $l-1$ in ρ''. Therefore, $\rho'' \models \neg C_l\varphi$, implying that $\mathcal{F}$ is not ck-informative.

The case in which $S2$ holds is treated similarly. ▮

Thus, we have proved:

Theorem A (weakly) information symmetric protocol $\mathcal{F}$ is ck-informative iff $\mathcal{F}$ is an $\mathcal{RCP}$.

E The Proof of Theorem 2

E.1 Part 1

We prove the first part of theorem 2 by the following sequence of lemmas:

Lemma 7 *Assume that the processor p_{i_k} transmits at l the atom*

$$a = \langle p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}, \alpha \rangle$$

in a run ρ of NIP where it is healthy at l. Then $M[\rho](p_{i_j}, p_{i_k}, l-1) = \emptyset$ for all $j = 1, \ldots, k-2$.

Proof: We prove the lemma by induction on the number of processors k appearing in the chain of the atom transmitted. For $k = 1$ and $k = 2$ the lemma is trivial.

Assuming that the lemma is correct for chains with less than k processors, we prove its correctness for chains with k processors. Assume by contradiction that for some j, $M[\rho](p_{i_j}, p_{i_k}, l-1) \neq \emptyset$, where $1 \leq j \leq k-2$, and let j^* be the biggest such j. We will see in the following lemma that the assertion $M[\rho](p_{i_{j^*}}, p_{i_k}, l-1) \neq \emptyset$ together with the consistency test in NIP, allow very little freedom to what any processor $p_{i_{j-1}}$, for $j^* < j \leq k-1$, may convey to any other processor at $l-k+j-1$ about the content of the atom $\langle p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-k+j-2}{\rightarrow} p_{i_{j-1}}, \alpha \rangle$ without being detected ill by the receiving processor.

To make this claim more precise let $b = \langle p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-k+j-1}{\rightarrow} p_{i_j}, \alpha \rangle$, for $j^* < j \leq k$, and consider the four one-parameter predicates φ_i, for $i = 1, \ldots, 4$:

- $\rho \models \varphi_1(b)$ iff p_{i_j} is healthy at $l-k+j-1$ in ρ, and $p_{i_{j-1}}$ conveys to p_{i_j} at that round the atom
$$\langle p_{i_1} \rightarrow \ldots \overset{l-k+j-2}{\rightarrow} p_{i_{j-1}}, \alpha \rangle.$$
- $\rho \models \varphi_2(b)$ iff p_{i_j} is healthy at $l-k+j-1$ in ρ, and $p_{i_{j-1}}$ conveys to p_{i_j} at that round the atom
$$\langle p_{i_f} \rightarrow \ldots \overset{l-k+j-2}{\rightarrow} p_{i_{j-1}}, \beta \rangle$$
where $j^* < f < j-1$, and $\beta =$ "detected ill" or $\beta = \emptyset$.
- $\rho \models \varphi_3(b)$ iff p_{i_j} is healthy at $l-k+j-1$ in ρ and p_{i_j} either detects at $l-k+j-1$ in ρ that $p_{i_{j-1}}$ was ill at $l-k+j-1$ or
$$M[\rho](p_{i_{j-1}}, p_{i_j}, l-k+j-1) = \emptyset.$$
- $\rho \models \varphi_4(b)$ iff p_{i_j} is not healthy at $l-k+j-1$ in ρ.

Finally, let $\varphi = \bigvee \varphi_i$. The following lemma specifies some of p_{i_k}'s knowledge at $l-1$ in ρ if $p_{i_{j^*}}$ transmits to it at $l-1$ a non-empty message.

Lemma 8 *Assume that p_{i_k} is healthy at $l-1$ in ρ. Let $M[\rho](p_{i_{j^*}}, p_{i_k}, l-1) \neq \emptyset$ for some j^*, $1 \leq j^* \leq k-2$, and assume that $p_{i_{j^*}}$ conveys to p_{i_k} at $l-k+j^*$ the atom*
$$\langle p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-k+j^*-1}{\rightarrow} p_{i_{j^*}}, \alpha \rangle.$$

Then for any processor p_{i_j}, $j^* < j \leq k-1$,
$$\rho \models \varphi(\langle p_{i_1} \rightarrow \ldots \overset{l-k+j-1}{\rightarrow} p_{i_j}, \alpha \rangle)$$
and
$$\rho \models \varphi(\langle p_{i_1} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}, \alpha \rangle).$$

Proof: We prove this lemma by induction on $k - j^*$, the length of the subchain $p_{i_{j^*+1}} \rightarrow \ldots \rightarrow p_{i_k}$.

Base $k = j^* + 1$: Here we need only show
$$\rho \models \varphi(\langle p_{i_1} \rightarrow \ldots \rightarrow p_{i_{j^*}} \overset{l-k+j^*}{\rightarrow} p_{i_{j^*+1}}, \alpha \rangle).$$

This holds by assumption.

Inductive step: Assume that the lemma is correct for numbers smaller than $k - j^*$. Here is the proof for $k - j^*$. If $p_{i_{k-1}}$ is not healthy at $l - 2$ in ρ, then

$$\rho \models \varphi_4(\langle p_{i_1} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}, \alpha\rangle)$$

and

$$\rho \models \varphi_3(\langle p_{i_1} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}, \alpha\rangle).$$

Otherwise, let $p_{i_{k-1}}$ be healthy at $l - 2$ in ρ. Since $M[\rho](p_{i_{j^*}}, p_{i_k}, l - 1) \neq \emptyset$, also $M[\rho](p_{i_{j^*}}, p_{i_{k-1}}, l - 2) \neq \emptyset$. By the inductive hypothesis, for all p_{i_j} and $j^* < j < k - 1$,

$$\rho \models \varphi(\langle p_{i_1} \rightarrow \ldots \overset{l-k+j-1}{\rightarrow} p_{i_j}, \alpha\rangle)$$

and

$$\rho \models \varphi(\langle p_{i_1} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}, \alpha\rangle).$$

We proceed to show

$$\rho \models \varphi(a)$$

where as above, $a = \langle p_{i_1} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}, \alpha\rangle$.

First, by assumption, $\rho \not\models \varphi_4(a)$. Second, p_{i_k} discovers at $l - 1$ in ρ that $p_{i_{k-1}}$ is ill at $l - 1$ or that $M[\rho](p_{i_{k-1}}, p_{i_k}, l - 1) = \emptyset$ iff $\rho \models \varphi_3(a)$. Assume next that $\rho \not\models \varphi_3(a)$. Thus,

$$(p_{i_{k-1}}, l - 1) \in \text{nc-pseudo-healthy}[\rho](p_{i_k}, l - 1).$$

Applying the inductive hypothesis, $\rho \models \varphi(a')$, where

$$a' = \langle p_{i_1} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}, \alpha\rangle.$$

Thus, $\rho \models \bigvee \varphi_i(a')$. Clearly, $\rho \not\models \varphi_4(a')$. Consider first the case where $p_{i_{k-1}}$ is healthy at $l - 1$ in ρ. If $\rho \models \varphi_1(a')$, then $\rho \models \varphi_1(a)$. Similarly, $\rho \models \varphi_2(a')$ implies that $\rho \models \varphi_2(a)$. Finally, if $\rho \models \varphi_3(a')$, then $\rho \models \varphi_2(a)$. Thus, since $\rho \models \varphi(a')$ we have $\rho \models \varphi(a)$.

Next assume that $p_{i_{k-1}}$ is ill at $l - 1$ in ρ. Since NIP is ck-informative and by assumption $(p_{i_{k-1}}, l - 1) \in$ nc-pseudo-healthy$[\rho](p_{i_k}, l - 1)$, $p_{i_{k-1}}$ must have conveyed to p_{i_k} at $l - 1$ in ρ one of the following two atoms:

1. $\langle p_{i_1} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}, \gamma\rangle$.

2. $\langle p_{i_f} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}, \beta\rangle$, where $j* < f < k-1$ and $\beta =$ "detected ill" or $\beta = \emptyset$.

In the first case, if $\gamma \neq \alpha$, then $M[\rho](p_{i_{k-1}}, p_{i_k}, l-1)$ will fail the consistency test that p_{i_k} runs at $l-1$ in ρ. Since this contradicts $(p_{i_{k-1}}, l-1) \in$ nc-pseudo-healthy$[\rho](p_{i_k}, l-1)$, $\gamma = \alpha$; thus $\rho \models \varphi_1(a)$. The second case implies $\rho \models \varphi_2(a)$. Therefore, $\rho \models \varphi(a)$. ∎

We now continue the proof of lemma 7. By assumption, p_{i_k} transmits at l the atom $a = \langle p_{i_1} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}, \alpha\rangle$, thus,

$$(p_{i_{k-1}}, l-1) \in \text{ nc-pseudo-healthy}[\rho](p_{i_k}, l-1).$$

By lemma 8 and the consistency test in NIP, $p_{i_{j*}}$ must have conveyed to p_{i_k} at $l-k+j^*$ the atom

$$\langle p_{i_1} \rightarrow \ldots \overset{l-k+j^*-1}{\rightarrow} p_{i_{j*}}, \alpha\rangle.$$

Clearly, $M[\rho](p_{i_{j*}}, p_{i_k}, l-1) \neq \emptyset$ implies that $M[\rho](p_{i_{j*}}, p_{i_k}, l-2) \neq \emptyset$. Thus, by lemma 8, $\rho \models \varphi(\tilde{a})$, where

$$\tilde{a} = \langle p_{i_1} \rightarrow \ldots \rightarrow p_{i_{k-2}} \overset{l-2}{\rightarrow} p_{i_k}, \alpha\rangle.$$

We consider the following three cases: First, if $\rho \models \varphi_1(\tilde{a})$, then recalling how Δ is evaluated, p_{i_k} should not have transmitted the atom a at l in ρ – a contradiction.

Second, assume that $\rho \models \varphi_2(\tilde{a})$. Recall that

$$(p_{i_k}, l-1) \in \text{ nc-pseudo-healthy}(p_{i_k}, l-1),$$

and $p_{i_{k-1}}$ conveys to p_{i_k} at $l-1$ the atom

$$b = \langle p_{i_1} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}, \alpha\rangle.$$

We distinguish between two cases: Assume first that $p_{i_{k-1}}$ conveys to p_{i_k} at $l-1$ that $p_{i_{k-2}}$ *transmitted* to it the atom

$$c = \langle p_{i_1} \rightarrow \ldots \overset{l-3}{\rightarrow} p_{i_{k-2}}, \alpha\rangle.$$

Then by the inductive hypothesis, $p_{i_{k-1}}$ must have conveyed to p_{i_k} at $l-1$ that $p_{i_{k-2}}$ conveyed to it that $M(p_{i_j}, p_{i_{k-2}}, l-3) = \emptyset$ for $j = 1, \ldots, k-4$. Since p_{i_k} transmits a at l, $p_{i_{k-1}}$ must have conveyed to p_{i_k} at $l-1$ that

$$M(p_{i_{k-3}}, p_{i_{k-1}}, l-2) = \emptyset,$$

and for the same reason, $M(p_{i_{k-2}}, p_{i_k}, l-1) = \emptyset$. Thus, $M(p_{i_j}, p_{i_k}, l-1) = \emptyset$ for $j = 1, \ldots, k-2$. This contradicts the existence of j^*.

Assume next that $p_{i_{k-1}}$ conveys to p_{i_k} at $l-1$ that $p_{i_{k-2}}$ conveyed but did not transmit the atom c to it. Then recalling how Δ is evaluated, p_{i_k} should not have transmitted a at l in ρ – a contradiction.

Finally, the case where $\rho \models \varphi_3(\tilde{a})$ follows similarly. This completes the proof. ▮

Corollary 2 *Assume that the processor p_{i_k} transmits at l the chain*

$$p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \xrightarrow{l-1} p_{i_k}$$

in a run of NIP where it is healthy at l. Then $k \leq t+1$.

Proof: By lemma 7, for $j = 1, \ldots, k-2$, $M(p_{i_j}, p_{i_k}, l-1) = \emptyset$. Clearly $k-2 \leq t$, but we argue that in fact $k-2 < t$. Indeed, had p_{i_k} seen precisely t processors failing at $l-1$, it should not have transmitted anything besides its $INPUT$ at that round. Thus, $k \leq t+1$. ▮

Corollary 3 *Each processor that is healthy in a run of NIP uses at most*

$$(t+1)\log n$$

bits for transmitting a chain.

Proof: The processor uses the trivial binary encoding of that chain. ▮

Lemma 9 *If the processor p_{i_k} transmits at l the chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \xrightarrow{l-1} p_{i_k}$ in a run ρ of NIP where it is healthy at l, for $k > 2$, then at least one of the following three must have occurred:*

1. *$p_{i_{k-1}}$ conveys an actual lie with chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \xrightarrow{l-2} p_{i_{k-1}}$ to p_{i_k} at $l-1$ in ρ.*
2. *$p_{i_{k-2}}$ conveys an actual lie with chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \xrightarrow{l-3} p_{i_{k-2}}$ to $p_{i_{k-1}}$ at $l-2$ in ρ.*
3. *$p_{i_{k-2}}$ conveys an actual lie with chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \xrightarrow{l-3} p_{i_{k-2}}$ to p_{i_k} at $l-2$ in ρ.*

Proof: Assume by contradiction that none of the above hold. Then the chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-3}{\rightarrow} p_{i_{k-2}}$ that $p_{i_{k-2}}$ conveys at $l-2$ carries exactly the same content to both $p_{i_{k-1}}$ and p_{i_k}. Moreover, the chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}$ that $p_{i_{k-1}}$ conveys to p_{i_k} at $l-1$ also carries that content. Finally, since p_{i_k} follows NIP at l, it will not transmit the chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}$ – a contradiction. ▮

We now prove the first claim of theorem 2. The idea is to pick, one at a time, each chain consisting of at least three processors that p_{i_k} transmits at l, and to *mark* the last actual lie performed on that chain. Lemma 9 implies that the only actual lies that will be marked are the ones perpetrated in rounds $l-1$ and $l-2$. A more careful examination shows that each actual lie at $l-1$ is marked at most once, and that each actual lie at $l-2$ is marked at most $n-2$ times.

More precisely, let p_{i_k} be healthy at l and assume that it transmits the chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-1}{\rightarrow} p_{i_k}$ at l, where $k \geq 3$. Lemma 9 motivates the introduction of the following *marking of actual lies.* If $p_{i_{k-1}}$ conveys an actual lie with chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}$ to p_{i_k} at $l-1$, then mark it; this actual lie is of *type 1.* Otherwise, if $p_{i_{k-2}}$ conveys an actual lie with chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-3}{\rightarrow} p_{i_{k-2}}$ to $p_{i_{k-1}}$ at $l-2$, mark it, and refer to it as *type 2.* Finally, if none of the above hold, mark the actual lie corresponding to the chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-3}{\rightarrow} p_{i_{k-2}}$ that $p_{i_{k-2}}$ must have conveyed to p_{i_k} at $l-2$ and call it *type 3.*

We now prove the following three lemmas referring to actual lies of types 1, 2 and 3:

Lemma 10 *Every actual lie of type 1 is marked at most once.*

Proof: For every actual lie of type 1 with chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-2}{\rightarrow} p_{i_{k-1}}$, there exists at most one chain, $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \rightarrow p_{i_{k-1}} \overset{l-1}{\rightarrow} p_{i_k}$, that p_{i_k} may transmit at l. ▮

Lemma 11 *Every actual lie of type 2 is marked at most once.*

Proof: Similar to lemma 10. ▮

Lemma 12 *Every actual lie of type 3 is marked at most $n-2$ times.*

Proof: For every actual lie of type 3 with chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \overset{l-3}{\rightarrow} p_{i_{k-2}}$, there exists at most $n-2$ chains, $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \rightarrow p_{i_{k-2}} \rightarrow q \overset{l-1}{\rightarrow} p_{i_k}$, $q \neq p_{i_{k-2}}$ and $q \neq p_{i_k}$ that p_{i_k} may transmit at l. ▮

And now the statement and proof of the first claim of theorem 2:

Corollary 4 *The number of bits that the processor p transmits at round l to another processor in a run ρ of NIP where it is healthy is less than*

$$n((t+1)\log n + |\rho|)AL^{+}[\rho](l-2, l-1).$$

Proof: It follows from the discussion above tha~~e~~t p transmits at l less than

$$nAL[\rho](l-2, l-1)$$

atoms whose chains consist of at least three processors. p also transmits less than n atoms whose chains consist of precisely two processors, and one atom whose chain consists of a single processor, namely p.

Thus, p transmits less than

$$nAL^{+}[\rho](l-2, l-1) = nAL[\rho](l-2, l-1) + (n-1) + 1$$

atoms. Now, by corollary 3, each atom that p transmits at l requires at most

$$(t+1)\log n + |\rho|$$

bits. ▮

E.2 Part 2

We now prove the second part of theorem 2 which states that:

Lemma 13 *The time needed for calculating the messages that p transmits at l in ρ using the routines described in appendix A is*

$$cAL^{+}[\rho](l-3, l-1)$$

where $c = poly(n, t, |\rho|)$. The space used in that calculation is

$$c'AL^{+}[\rho](l-t-1, l-1)$$

where, as before, $c' = poly(n, t, |\rho|)$.

Before proving this statement we need the following two lemmas:

Lemma 14 *Let p be healthy at round l in a run of NIP. The sum of the sizes of the messages that p receives at l is bounded by $AL^+(l-2,l)$ times a low degree polynomial in n, t and $|\rho|$.*

Proof: Let $L(q,p,l)$ denote the number of actual lies that q conveys to p at l. Let $RD(p,l)$ denote the number of actual lies that were conveyed to p at l; thus $RD(p,l) = \sum_q L(q,p,l)$. Let μ be a bound on the number of bits that are required in order to represent any atom in NIP. By lemma 3 we may let

$$\mu = (t+1)\log n \overset{t}{\cancel{+}} |\rho|.$$

Using the estimates in appendix E.1

$$\begin{aligned}|M(q,p,l)| \;\leq\; & \mu\Big\{L(q,p,l) + \sum_r L(r,q,l-1) + (n-2)\sum_s L(s,q,l-2) \\ & + \sum_{s,r} L(s,r,l-2) + n\Big\}\end{aligned}$$

Summing over q, $q \neq p$,

$$\begin{aligned}\sum_q |M(q,p,l)| \;\leq\; & \mu\Big\{\sum_q L(q,p,l) + \sum_{q,r} L(r,q,l-1) + (n-2)\sum_{q,s} L(s,q,l-2) \\ & + \sum_{q,s,r} L(s,r,l-2) + n^2\Big\} \\ \leq\; & \mu\left\{RD(p,l) + AL(l-1,l-1) + (n-2)AL(l-2,l-2)\right. \\ & \left. + (n-1)AL(l-2,l-2) + n^2\right\} \\ \leq\; & n^2\mu AL^+(l-2,l) \qquad \blacksquare\end{aligned}$$

Lemma 15 *Let p be healthy at round l in a run of NIP. The size of*

$$\cup_r LB(r,p,l-1)$$

is bounded by $AL^+(l-2,l-1)$ times a low degree polynomial in n, t and $|\rho|$.

Proof: Following the notations of lemma 14 we argue that

$$|LB(r,p,l-1)| \leq \mu\left\{L(r,p,l-1)+\sum_{s} L(s,p,l-2)+\sum_{s} L(s,r,l-2)+1\right\}.$$

Summing over r, $r \neq p$,

$$\begin{aligned}
|\cup_r LB(r,p,l-1)| &\leq \mu\left\{\sum_{r} L(r,p,l-1)+\sum_{r,s} L(s,p,l-2)\right. \\
&\qquad \left. +\sum_{r,s} L(s,r,l-2)+n\right\} \\
&\leq \mu\{RD(p,l-1)+(n-1)RD(p,l-2) \\
&\qquad +AL(l-2,l-2)+n\} \\
&\leq n\mu AL^{+}(l-2,l-1) \qquad \blacksquare
\end{aligned}$$

We proceed to prove lemma 13.

Proof: We examine bottom up the complexity of each of the subroutines in NIP. To this end consider a processor p that is healthy at round l in a run of NIP. We estimate the time it spends in the different routines.

Subroutine DETECT_ILL is $\mathcal{O}(1)$ and subroutine CHECK_ILL is $\mathcal{O}(n)$. Invoking TRUST is $\mathcal{O}(n)$ provided that no recursive calls to TRUST are made within it. Since there may be no more than t such calls, the complexity of TRUST is $\mathcal{O}(nt)$.

The total time that p spends in routine NEW_INFORMATION is bounded by the size of the messages that it receives at l times a low degree polynomial in n and t.

Evaluating the time that p spends in CONSISTENT is a little more involved. At any given call to CONSISTENT, there are at most n invocations to TRUST, which is $\mathcal{O}(n^2t)$, and the time spent at step CHECK is $\mathcal{O}(t^2)$. Now when CONSISTENT is invoked with parameters r, k and $TREE$, the number of recursive calls to CONSISTENT is no more that the number of internal nodes in $TREE$.

Thus the total time that p spends in CONSISTENT is bounded by the the sum of the sizes of the messages that it receives at l, times a low degree polynomial in n and t.

Consider now routine SET_LABEL. The main computational effort in that routine is spent evaluating the formula

$$\{ST(s,r,l-2|M(q,p,l)) \bigtriangledown LB(s,q,l-2)\} \bigtriangleup ST(s,r,l-2|LB(r,p,l-1))$$

for every s and r. We evaluate that formula by first scanning the transmission tree

$$ST(s,r,l-2|M(q,p,l))$$

and next the transmission tree

$$ST(s,r,l-2|LB(r,p,l-1)).$$

Thus the time spent in SET_LABEL(q,p,l) is bounded by the sizes of both $M(q,p,l)$ and $\cup_r LB(r,p,l-1)$ multiplied by a factor as above. Thus the total time spent in calls to SET_LABEL is bounded by the sum of the sizes of the messages that p receives at l plus n times the size of $\cup_r LB(r,p,l-1)$, with all this multiplied by a factor as above.

Therefore the total time that p spends at l in routine UPDATE_NK is bounded by the sum of sizes of the messages that p receives at l plus the size of $\cup_r LB(r,p,l-1)$ with all this multiplied by a low degree polynomial in n and t.

Now, by lemma 14 the sum of the sizes of the messages that p receives at l is bounded by $AL^+(l-2,l)$ times a low degree polynomial in n, t and $|\rho|$. By lemma 15 the size of $\cup_r LB(r,p,l-1)$ is bounded by $AL^+(l-2,l-1)$ times a low degree polynomial in n, t and $|\rho|$. This completes the first part of lemma 13.

The second part follows since the number of processors defining any chain that is forwarded by a healthy processor in NIP is no more than $t+1$. ∎

F The Proof of Theorem 3

In this appendix we prove theorem 3 which states:

Theorem NIP is a communication efficient ck-informative protocol.

Proof: Fix a run σ of NIP, a ck-informative protocol $\mathcal{F}$ and a round l. We will construct a run $\rho \in DOM(\mathcal{F},\sigma)$ so that

$$CC[\sigma](l) \leq p(n,t)CC[\rho](l).$$

In the process of doing so we will explicitly construct the polynomial $p(n,t)$. We assume hereafter that $\Sigma = \Delta = \{0,1\}$.

In order to simplify the construction of ρ, we first modify σ. Let r be ill at k and let q be healthy at $k+1$, both in σ. Suppose that q detects at k in σ that r is ill at k. Then we let r transmit to q at k in the modified run the message $M'(r,q,k)$ defined at the end of section 6.2. Thus, in this new run $(r,k) \in$ nc-pseudo-healthy(q,k), the messages that are transmitted at $k+1$ and $k+2$ are modified only slightly, and the messages that are transmitted at other rounds remain as in σ. Notice that each actual lie that r conveys to q at k in this modified run corresponds to an actual lie in σ. In ρ, we will let r convey these actual lies to q at k as well, so we may assume hereafter that σ itself satisfies the above.

We first construct an auxiliary run σ_x which is very similar to σ, but in which some atoms have the form $\langle ch, x_{ch}\rangle$, where x_{ch} is an undetermined content. x_{ch} will be assigned an element in $\{0,1\}^*$ when we construct ρ from σ_x.

Let the parameters n, t, $\mathcal{F}$, $\mathcal{O}$ and CA in σ_x be as in σ. Thus, for σ_x to be fully specified we need to choose an adversary and the inputs. In order to do so we sequentially scan the processors at rounds 1 to l as follows: We start with p_1 at round 1, then p_2 at round 1, up to p_n at 1. Next we scan p_1 at round 2, p_2 at round 2, etc.

Suppose that we are currently scanning processor q at round k. If q is either ill or dead at k in σ we proceed to its neighbor. Otherwise, q is healthy at k in σ. Consider each of the atoms $\langle ch, \alpha\rangle$ that q transmits there.

First, if $ch = \xrightarrow{k-1} q$, then let that chain carry the content x_{ch} in σ_x. Second, if $ch = p_{i_f} \xrightarrow{k-1} q$, p_{i_f} is ill at $k-1$, and q is the processor with smallest index that is healthy at k, then let that chain carry the content x_{ch} in σ_x. Third, assume that $ch = p_{i_1} \rightarrow \ldots \rightarrow p_{i_f} \xrightarrow{k-1} q$ and p_{i_f} conveyed the actual lie $\langle p_{i_1} \rightarrow \ldots \xrightarrow{k-2} p_{i_f}, \alpha'\rangle$ to some processor at $k-1$. Then let $ch_r = p_{i_1} \rightarrow \ldots \rightarrow p_{i_f} \xrightarrow{k-1} r$. If x_{ch_r} has not yet been assigned for any r, then let ch carry the content x_{ch} in σ_x.

Finally, assume that $ch = p_{i_1} \rightarrow \ldots \rightarrow p_{i_f} \xrightarrow{k-1} q$ and p_{i_f} did not convey any actual lie with chain $p_{i_1} \rightarrow \ldots \xrightarrow{k-1} p_{i_f}$ in σ. Thus, by lemma 9, $p_{i_{f-1}}$ must have conveyed an actual lie with chain $p_{i_1} \rightarrow \ldots \xrightarrow{k-2} p_{i_{f-1}}$. Let

$$\begin{aligned} ch_p &= p_{i_1} \rightarrow \ldots \rightarrow p_{i_{f-1}} \xrightarrow{k-2} p \\ ch_{s\rightarrow r} &= p_{i_1} \rightarrow \ldots \rightarrow p_{i_{f-1}} \rightarrow s \xrightarrow{k-1} r. \end{aligned}$$

If neither x_{ch_p} nor $x_{ch_{s\rightarrow r}}$ have been assigned for any p, or pair $s \rightarrow r$, then let ch carry the content x_{ch} in σ_x.

For each atom $\langle ch, x_{ch}\rangle$ in σ_x, let $|\langle ch, x_{ch}\rangle|_\infty$ be the ∞-length of the corresponding atom in σ. Let $X[\sigma_x](q,k)$ be the set of atoms $\langle ch, x_{ch}\rangle$ for $ch = p_{i_1} \rightarrow \ldots \rightarrow p_{i_f} \xrightarrow{k-1} q$. The motivation for this rather cumbersome definition is the following

Lemma 16

$$CC[\sigma](l) \leq \sum_{\substack{(q,k)\in\text{healthy}\\ 0<k\leq l}} \sum_{a\in X[\sigma_x](q,k)} n^3((t+1)\log n + |a|_\infty).$$

Proof: We prove that each actual lie with ∞-length λ can force the processors that are healthy in θ to transmit at most

$$n^3((t+1)\log n + \lambda)$$

bits.

Assume that p_{i_f} conveys at k in the run θ of NIP an actual lie with chain $p_{i_1} \rightarrow \ldots \overset{k-1}{\rightarrow} p_{i_f}$. Let q be one of at most $n - 1$ processors that are healthy at $k+1$ in θ. Then q might have to transmit at most $n-2$ atoms at $k+1$ in θ. Also, every processor r other than p_{i_f} might have to transmit $n-3$ atoms at $k+2$ with chain $p_{i_1} \rightarrow \ldots \rightarrow p_{i_f} \rightarrow s \overset{k+1}{\rightarrow} r$, where $s \neq p_{i_f}, r$. Interestingly, processors that are healthy at rounds succeeding $k+2$ will not transmit any atom due to that actual lie.

Since

$$(n-1)(n-2) + (n-1)(n-2)(n-3) < n^3$$

this actual lie can force the transmission of no more than n^3 atoms, each of which involves no more than $(t+1)\log n + \lambda$ bits. ▮

We proceed to construct the run ρ. Let the parameters n, t, $\mathcal{F}$, $\mathcal{O}$ and CA in ρ be as in σ_x, and therefore also as in σ. Let

$$|X[\sigma_x](q,k)|_\infty = \sum_{a\in X[\sigma_x](q,k)} |a|_\infty.$$

Our goal is to force q to transmit at k in ρ at least $|X[\sigma_x](q,k)|_\infty$ bits to some other processor, say r, by carefully selecting in ρ a new short content for each of the atoms in $X[\sigma_x](q,k)$. In order to do so we use the natural one-to-one mapping from atoms in $X[\sigma_x](q,k)$ into actual lies in σ, and we assign in ρ a new content to each of these actual lies. Thus, the run ρ that we construct in this way is in $DOM(\mathcal{F},\sigma)$. Suppose that we had satisfied the above. Then, recalling that $|a|_\infty \geq 1$, we could argue the following:

$$CC[\sigma](l) \leq \sum_{\substack{(q,k)\in\text{healthy}\\ 0<k\leq l}} \sum_{a\in X[\sigma_x](q,k)} n^3((t+1)\log n + |a|_\infty)$$

$$\begin{aligned}
&\leq \quad n^3((t+1)\log n+1) \sum_{\substack{(q,k)\in\text{healthy}\\0<k\leq l}} \; \sum_{a\in X[\sigma_x](q,k)} |a|_\infty \\
&\leq \quad n^3((t+1)\log n+1) \sum_{\substack{(q,k)\in\text{healthy}\\0<k\leq l}} |X[\sigma_x](q,k)|_\infty \\
&\leq \quad n^3((t+1)\log n+1) \sum_{\substack{(q,k)\in\text{healthy}\\r,\ 0<k\leq l}} |M[\rho](q,r,k)| \\
&= \quad n^3((t+1)\log n+1)CC[\rho](l).
\end{aligned}$$

Therefore, we would only have to choose a polynomial $p(n,t)$ so that $p(n,t) \geq n^3((t+1)\log n+1)$ in order to prove the theorem.

So we are left with the problem of letting q transmit at least $|X[\sigma_x](q,k)|_\infty$ bits to r at k in ρ, where we are assuming of course that $|X[\sigma_x](q,k)|_\infty > 0$. The idea is again to pick carefully a new content for each of the actual lies corresponding to the atoms in $X[\sigma_x](q,k)$, without disclosing thereby to q at $k-1$ in ρ that the sending processor was ill. An application of the pigeonhole principle will then force q to transmit at least $|X[\sigma_x](q,k)|_\infty$ bits at k in ρ as required.

Let $\langle ch, x_{ch}\rangle \in X[\sigma_x](q,k)$ for $ch = p_{i_1} \rightarrow \ldots \rightarrow p_{i_f} \overset{k-1}{\rightarrow} q$, and let $\langle ch, \alpha\rangle$ be the corresponding atom that q transmits in σ. Let $\alpha \neq \emptyset$, and consider the more interesting case where $f > 1$. Assume first that p_{i_f} conveyed an actual lie with chain $p_{i_1} \rightarrow \ldots \overset{k-2}{\rightarrow} p_{i_f}$. Then it is a property of the consistency test in NIP that q would not be able to determine at $k-1$ that p_{i_f} was ill at $k-1$, had p_{i_f} instead conveyed to q the atom $\langle p_{i_1} \rightarrow \ldots \overset{k-2}{\rightarrow} p_{i_f}, \beta\rangle$, where $\beta \neq \alpha, \emptyset$, "detected ill".

Assume now that p_{i_f} was healthy at $k-1$. Then $p_{i_{f-1}}$ must have conveyed an actual lie with chain $p_{i_1} \rightarrow \ldots \overset{k-3}{\rightarrow} p_{i_{f-1}}$. Again it is a property of the consistency test in NIP that p_{i_f} would not be able to determine at $k-1$ that $p_{i_{f-1}}$ was ill at $k-2$, had $p_{i_{f-1}}$ instead conveyed to it the atom $\langle p_{i_1} \rightarrow \ldots \overset{k-3}{\rightarrow} p_{i_{f-1}}, \beta\rangle$, where $\beta \neq \alpha, \emptyset$, "detected ill". Finally, if p_{i_f} was ill at $k-1$ but did not convey in σ any actual lie with chain $p_{i_1} \rightarrow \ldots \overset{k-2}{\rightarrow} p_{i_f}$, then if $p_{i_{f-1}}$ conveys to p_{i_f} the atom $\langle p_{i_1} \rightarrow \ldots \overset{k-3}{\rightarrow} p_{i_{f-1}}, \beta\rangle$, with β as above, and p_{i_f} forwards it to q at $k-1$, then q would not be able to determine that p_{i_f} was ill at $k-1$. Further, the above holds also if we modify the content of several actual lies simultaneously.

The case where $\alpha = \emptyset$ is treated similarly. Notice that if q transmits the atom $\langle p_{i_1} \rightarrow \ldots \rightarrow p_{i_f} \overset{k-1}{\rightarrow} q, \emptyset\rangle$ at k in σ, then it does not transmit any atom with chain

$$\ldots \rightarrow p_{i_1} \rightarrow s \rightarrow p_{i_2} \rightarrow \ldots \rightarrow p_{i_f} \overset{k-1}{\rightarrow} q.$$

Otherwise, the former atom would be implicit in the transmission of the latter. As explained above, we may let q receive the atom $\langle p_{i_1} \rightarrow \ldots \overset{k-2}{\rightarrow} p_{i_f}, \beta\rangle$ without creating thereby any inconsistency in the message that p_{i_f} transmits to q at $k-1$. To see this, recall that p_{i_f} did not convey to q at $k-1$ in σ any atom with chain

$$\ldots \rightarrow p_{i_1} \rightarrow s \rightarrow p_{i_2} \rightarrow \ldots \overset{k-2}{\rightarrow} p_{i_f}$$

carrying new information to q. Thus p_{i_1} could certainly have transmitted to p_{i_2} at $k-f$ in σ without creating any such inconsistency. Furthermore, we are not introducing any new actual lie.

Notice that we are implicitly assuming that a processor r that is ill at round k might choose not to convey that it detected at $k-1$ that some other processor p was ill at $k-1$. Instead, r may selectively forward some of the atoms it received from p at $k-1$ to processors at k.

Finally we apply the pigeonhole principle in order to select a new content in ρ for each of the atoms in $X[\sigma_x](q,k)$. Let τ stand for $|X[\sigma_x](q,k)|_\infty$. First, notice that there are $2^\tau - 2$ messages that are strictly less than τ bits long, excluding of course the empty message which is never transmitted by a processor when it is healthy. Second, for every atom $a \in X[\sigma_x](q,k)$ there are $2^{|a|_\infty+1} - 1$ different atoms a' carrying the same chain as a and satisfying $|a'| \leq |a|_\infty$. But

$$\prod_a (2^{|a|_\infty+1} - 1) \geq \prod_a 2^{|a|_\infty} = 2^{\sum_a |a|_\infty} > 2^\tau - 2,$$

where a is an arbitrary atom in $X[\sigma_x](q,k)$. Thus there exists at least one choice of content for the atoms in $X[\sigma_x](q,k)$ that will make q transmit at least $\tau = |X[\sigma_x](q,k)|_\infty$ bits to r at k in ρ. This completes the proof. ▮

G The Proof of Theorem 4

In this appendix we prove theorem 4 which states:

Theorem For every n and t and for every ck-informative protocol with these parameters, there exists a run ρ of that protocol with $|\rho| = 1$ in which some processor transmits at least c^t bits at a round in which it is healthy, for $c > 1$.

Proof: By virtue of theorem 3, it is sufficient to build a run of NIP in which some processor transmits at least c^t bits at a round in which it is healthy.

We build the following run: Let $f = \lfloor (t-1)/2 \rfloor$. Let p_{2i-1} and p_{2i} be ill at i, for $i = 1, \ldots, f$, but let them follow NIP there. p_{2f+1} will also be ill at $f+1$ and it will also essentially follow NIP, but it will also transmit some (and in fact many) actual lies. p_{2f+1} will forge an actual lie at $f+1$ for every chain $p_{i_1} \rightarrow p_{i_2} \rightarrow \ldots \xrightarrow{f} p_{i_f}$, satisfying either $p_{i_j} = p_{2j-1}$ or $p_{i_j} = p_{2j}$, for every $j = 1, \ldots, f$. Of course, there are exponentially (in t) many such chains.

For $k > 2f+1$ and every l, let p_k be healthy at l. It follows that $(p_{2f+1}, f+1) \in$ nc-pseudo-healthy$(p_k, f+1)$. Therefore for some choice of content for these actual lies each p_k will have to transmit at $f+2$ exponentially long messages in order to convey the message it received from p_{2f+1} at $f+1$. ▮

H The Proof of Theorem 5

In this appendix we sketch the proof of theorem 5 which states:

Theorem For every n and t and for every sba-informative protocol with these parameters, there exists a run of that protocol in which some processor transmits at least c^t bits at a round in which it is healthy, for $c > 1$.

Proof: (sketch) Consider a processor p that is healthy at round l in a run ρ of an sba-informative protocol. We argue that p must convey at l in ρ its reduced view if the following situation holds: If p does *not* convey at l in ρ its reduced view, then there exists a run ρ' that is (p, l)-weakly-equivalent to ρ, so that $CR[\rho'](l) = 0$.

Thus, in that case the notions of ck-informative and sba-informative coincide. Finally, we argue that the segment in the proof of theorem 4 satisfies this property. ▮

DISTRIBUTION LIST

Office of Naval Research Contract N00014-82-K-0154
Michael J. Fischer, Principal Investigator

Defense Technical Information Center
Building 5, Cameron Station
Alexandria, VA 22314
(12 copies)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. R.B. Grafton, Scientific
Officer (1 copy)

Information Systems Program (437)
(2 copies)

Code 200 (1 copy)
Code 455 (1 copy)
Code 458 (1 copy)

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, CA 91106
(1 copy)

Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375
(6 copies)

Office of Naval Research
Resident Representative
715 Broadway, 5th Floor
New York, NY 10003
(1 copy)

Dr. A.L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380
(1 copy)

Naval Ocean Systems Center
Advanced Software Technology Division
Code 5200
San Diego, CA 92152
(1 copy)

Mr. E.H. Gleissner
Naval Ship Research and Development Center
Computation and Mathematics Department
Bethesda, MD 20084
(1 copy)

Captain Grace M. Hopper
Naval Data Automation Command
Washington Navy Yard
Building 166
Washington, D.C. 20374
(1 copy)

Defense Advance Research Projects Agency
ATTN: Program Management/MIS
1400 Wilson Boulevard
Arlington, VA 22209
(3 copies)