

Easy Impossibility Proofs for Distributed Consensus Problems¹

Michael J. Fischer

Department of Computer Science
Yale University
P.O. Box 2158,
New Haven, Connecticut 06520 USA

Nancy A. Lynch

Laboratory for Computer Science
Massachusetts Institute of Technology
545 Technology Square
Cambridge, Massachusetts 02139, USA

Michael Merritt

AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, New Jersey 07974 USA

Abstract

Easy proofs are given of the impossibility of solving several consensus problems (Byzantine agreement, weak agreement, Byzantine firing squad, approximate agreement and clock synchronization) in certain communication graphs. It is shown that, in the presence of f faults, no solution to these problems exists for communication graphs with fewer than $3f+1$ nodes or less than $2f+1$ connectivity. While some of these results had been proven previously, the new proofs are much simpler, provide considerably more insight, apply to more general models of computation, and (particularly in the case of clock synchronization) significantly strengthen the results.

1. Introduction

In this paper we present easy proofs for the impossibility of solving several consensus problems in particular communication graphs. We prove results for Byzantine agreement, weak agreement, the Byzantine firing squad problem, approximate agreement and clock synchronization. The bounds are all the same: tolerating f faults requires at least $3m+1$ nodes and at least $2f+1$ connectivity in the communication graph. (The connectivity of a graph is the minimum number of nodes whose removal disconnects the graph. Also, we assume throughout that graphs have at least three nodes.) For a given value of f , we call graphs with fewer than $3f+1$ nodes or less than $2f+1$ connectivity *inadequate graphs*.

Each of our proofs is an argument by contradiction. We assume that a given problem can be

¹Earlier versions of this paper appeared in the ACM Conference Proceedings of PODC 1985, and in **Distributed Computing**, volume 1 number 1, reprinted by permission.

solved in a system with an inadequate communication graph. We then construct a set of system behaviors all of which cannot satisfy the correctness conditions for the given problem, even though they are required to do so. Versions of many of the results with proofs of this same general form were already known. Our proofs differ from the earlier proofs in the technique we use to construct the set of behaviors. Our technique is simpler and applies to more general models of distributed computation.

For Byzantine agreement both bounds were already known [PSL,D]. The $3f+1$ node lower bound in [PSL] was proved only for a particular synchronous model of computation. Although carefully done, the proof is somewhat complicated and not as intuitive as one might like. In contrast, our proof is simple, transparent, and applies to general models of computation. A proof of the $2f+1$ connectivity lower bound was presented informally in [D]; we prove that bound more formally and for more general models.

For weak Byzantine agreement, the requirement of $3f+1$ nodes was known [L], but it was proved using a complicated construction. The new proof is easy and extends to more general models (although not as general as those for Byzantine agreement and approximate agreement). The $2f+1$ connectivity requirement was previously unknown. The result for the Byzantine firing squad problem follows from a reduction to weak agreement in [CDDS]. We provide a direct proof. For approximate agreement, the $3f+1$ bound was noted, but not proved, in [DLPSW], while the $2f+1$ connectivity requirement was previously unknown.

For clock synchronization, the $3f+1$ node bound was proved in [DHS] using a complicated proof. The authors of [DHS] claimed that they knew how to prove the corresponding $2f+1$ connectivity lower bound, but we presume that such a proof also would be complicated. We prove both the $3f+1$ node and the $2f+1$ connectivity bounds, for a much more general notion of clock synchronization than in [DHS]. These synchronization bounds assume that there is no direct method, other than by reading their inaccurate hardware clocks, for nodes to measure the passage of time.

Since we obtain the same lower bounds for each problem, one might think that the problems are equivalent in some sense. This is not the case. The bounds for the different problems require different assumptions about the underlying model. For example, the lower bounds for Byzantine and approximate agreement work with virtually any reasonable computational model, while the lower bound for weak agreement requires a special assumption, placing a bound on the rate of propagation of information through the system. The bound for clock synchronization requires a different assumption, about how devices can measure time. Many of the results are sensitive to small perturbations of the underlying assumptions, about such factors as communication delay or the behaviors of faulty nodes.

2. A Model of Distributed Systems

To make the impossibility results clear, concise, and general, we introduce a simple model of distributed systems.

A *communication graph* is a directed graph G with node set $nodes(G)$ and edge set $edges(G)$, such that the directed edges occur in pairs; $edge(u,v) \in edges(G)$ if and only if $(v,u) \in edges(G)$. (We consider a pair of directed edges rather than a single undirected edge to model the communication in each direction separately). We call the edge (u,v) an *outedge* of u , and an *inedge* of v . Let U be a subset of $nodes(G)$. Then the *subgraph* G_U induced by U is the graph containing all the nodes in U and all the edges between nodes in U . The *inedge border* of G_U is the set of edges from nodes outside U into U ; that is, $edges(G) \cap ((nodes(G) \setminus U) \times U)$.

A *system* G is a communication graph G with an assignment of a *device* and an *input* to each node of G . Devices are undefined primitive objects. The specific inputs we consider are either encodings of Booleans or real numbers or real-valued functions of time (e.g. local clocks). The particular type of input depends on the agreement problem addressed. If a node is assigned device A in system G , we say that the node *runs* A . A *subsystem* U of G is any subgraph G_U of G with the associated devices and inputs.

Every system G has a *system behavior*, E , which is a tuple containing a *behavior* of every node and edge in G . (We also describe E as a behavior of the communication graph G . Note that a system has exactly one behavior, while a graph may have several, depending on the devices and inputs assigned to the nodes.) The restriction of a system behavior E to the behaviors of the nodes and edges of a subgraph G_U of G is the *scenario* E_U of G_U in E .

For now, we take node and edge behaviors as primitives. In more concrete and familiar models, a node or edge behavior might be a finite or infinite sequence of states, or a mapping from the positive reals to some state set, denoting state as a function of time. (We use the latter interpretation for later results). Less familiar models might interpret behaviors as mappings from reals to states, or from transfinite ordinals to states. To obtain our first results, the precise interpretation of node and edge behaviors is unimportant. We need restrict our model only so that the following two axioms hold. (We assume these two axioms throughout the paper. Some of the later results require additional assumptions.)

Locality Axiom Let G and G' be systems with behaviors E and E' , respectively, and isomorphic subsystems U and U' , (with vertex sets U and U'). If the corresponding behaviors of the inedge borders of U and U' in E and E' are identical, then scenarios E_U and $E_{U'}$ are identical.

The Locality axiom says that communication takes place only over the edges of the communication graph. In particular, it expresses the following property: The only parameters affecting the behavior of any local portion of a system are the devices and inputs at each local node, together with any information incoming over edges from the remainder of the system. If

these parameters are the same in two behaviors, the local behaviors (scenarios) are the same. ²Clearly, if some such locality property does not hold, then agreement is trivially achievable by having devices read other device's inputs directly.

Fault Axiom Let A be any device. Let E_1, \dots, E_d be d edge behaviors, such that each E_i is the behavior of the i 'th outedge of a node running A in some system behavior E^i . Let U be any node with d outgoing edges $(u, v_1), \dots, (u, v_d)$. There is a device F such that in any system in which U runs F , the behavior of each outedge (u, v_i) is E_i .

In this case, we write $F_A(E_1, \dots, E_d)$ for F . This axiom expresses a powerful masquerading capability of failed devices. Any behavior exhibited by a device over different edges in different system behaviors can be exhibited by a failed device in a single system behavior. When this axiom is significantly weakened (say, by adding an unforgeable signature assumption), then consensus is possible [LSP, PSL].

For establishing the relevance of our impossibility results to more concrete models of distributed systems, it is sufficient to interpret our definitions in the particular model and then to demonstrate that the Locality and Fault axioms hold under the particular interpretation.

Our proofs utilize the graph-theoretic notion of a *covering*. For any graph G , let $neighbors = \{(u, V) \mid U \text{ is a node of } G \text{ and } V \text{ is the set of all nodes } v \text{ such that there is an edge from } v \text{ to } U \text{ in } G\}$. A graph S *covers* G if there is a mapping ϕ from the nodes of S to the nodes of G that preserves "neighbors." That is, if node U of S has d neighbors v_1, \dots, v_d , and $\phi(u) = w$ for a node w of G , then w has d neighbors x_1, \dots, x_d and $\phi(v_i) = x_i$ for $1 \leq i \leq d$. Under such a mapping, S looks locally like G .

Graph coverings play an important role in our understanding of the interaction of network topology and distributed computation. A discussion appears in [A], and indeed, some of our proofs are surprisingly similar to Angluin's. Similar techniques also appear in [IR], [B] and elsewhere.

3. Byzantine Agreement

We say that Byzantine agreement is possible in a graph G (with n nodes) if there exist n devices A_1, \dots, A_n (which we call *agreement devices*), with the following properties.

Each agreement device A_u takes a Boolean input and chooses 1 or 0 as a result. (To model choosing a result, assume there is a function CHOOSE from behaviors of nodes running agreement devices to the set $\{0, 1\}$.) A node U of G is *correct* in a behavior E of G if node U runs A_u in E . Any system behavior E of G in which at least $n - f$ nodes are correct is a *correct*

²For weak agreement and the firing squad problem, we need to extend this locality property to include time.

system behavior. Correct system behaviors must satisfy the following conditions.

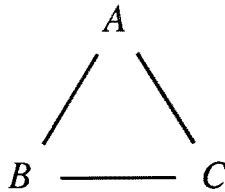
Agreement. Every correct node chooses the same value.

Validity. If all the correct nodes have the same input, that input must be the value chosen.

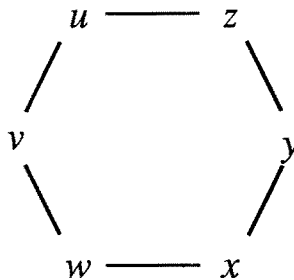
Theorem 1: *Byzantine agreement is not possible in inadequate graphs.*

3.1. Number of Nodes

We begin with the lower bound of $3f+1$ for the number of nodes required for Byzantine agreement. First consider the case where $|G| = n = 3$ and $f = 1$. Assume that the problem can be solved for the communication graph G consisting of three nodes fully connected by communication edges. Let the three nodes of G be a , b and c , and assume that they run agreement devices A , B and C , respectively. We represent each pair of directed edges by a single undirected edge, and label the nodes with the devices they run.

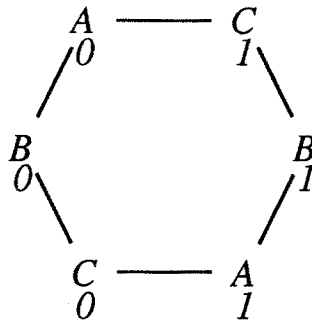


The covering graph S is as follows.



This graph looks locally like G under the mapping ϕ defined by $\phi(u) = \phi(x) = a$, $\phi(v) = \phi(y) = b$ and $\phi(w) = \phi(z) = c$.

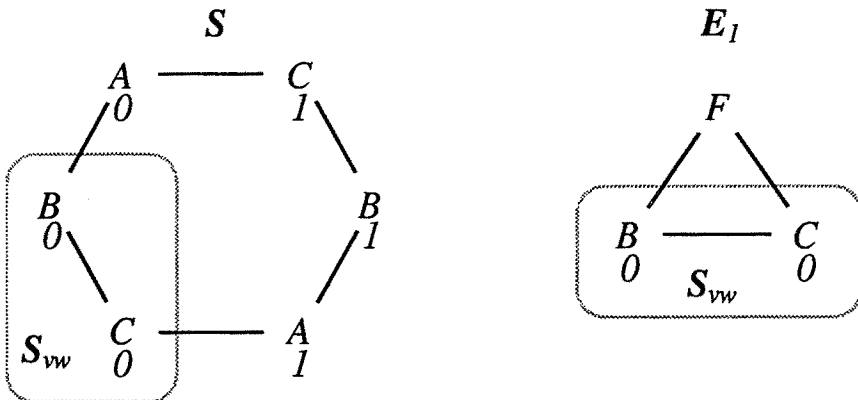
Now specify the system by assigning devices and inputs for the nodes in S as follows.



By this we mean that node u runs device A with input 0 , node v runs B with input 0 , and so on. Let S denote the resulting behavior of the system; S includes a behavior for each of the six nodes and twelve directed edges in S .

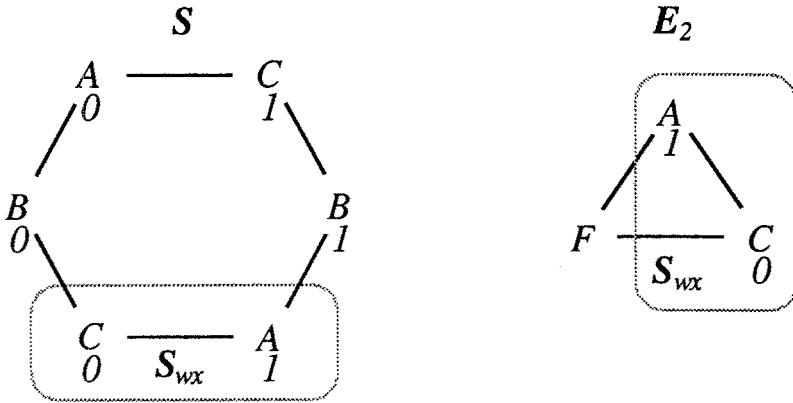
Now consider scenarios S_{vw} , S_{wx} and S_{xy} in S , where each consists of the behaviors of the two indicated nodes in S , along with the activity over the two connecting edges. We argue that each of these scenarios is identical to a scenario in a correct behavior of G .

The first scenario S_{vw} is shown below.



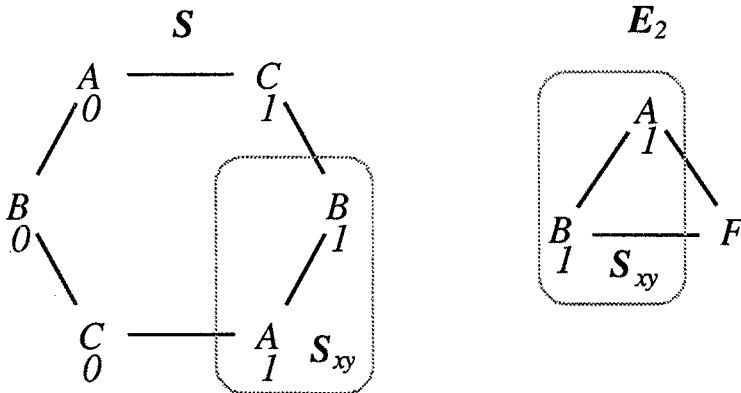
This scenario is the behavior in S of nodes v and w , together with that of the communication edges between v and w . Now consider the behavior E_1 of G in which node b runs B on input 0 , node c runs C on input 0 , and node a runs a device that mimics node u in talking to b , and mimics node x in talking to c . Formally, if $E_{(u,v)}$ and $E_{(x,w)}$ are the indicated edge behaviors in S , node a runs device $F_A(E_{(u,v)}, E_{(x,w)})$ (we have written just F in the figure). This device exists, by the Fault axiom, and in the resulting behavior, edges from node a to node b and to node c have behaviors $E_{(u,v)}$ and $E_{(x,w)}$, respectively. By the Locality axiom, the scenario containing b and c 's behaviors in E_1 is identical to S_{vw} . Validity requirements insure that node b and node c must choose 0 in E_1 . Since their behavior is identical in S , v and w choose 0 in S .

Next, consider scenario S_{wx}



This scenario includes the behavior of nodes w and x in S . It is also the behavior of nodes a and c in a behavior E_2 of G which results when they run their devices A and C on inputs 1 and 0, respectively, and node b is faulty, exhibiting the same behavior to node x that v exhibits to w in S , and the same behavior to node a that y exhibits to x in S . The behavior of node c in E_2 is identical to that of node w in S , so node c chooses 0 in E_2 , from the argument above. By agreement, node a decides 0 in E_2 . Thus node x decides 0 in S .

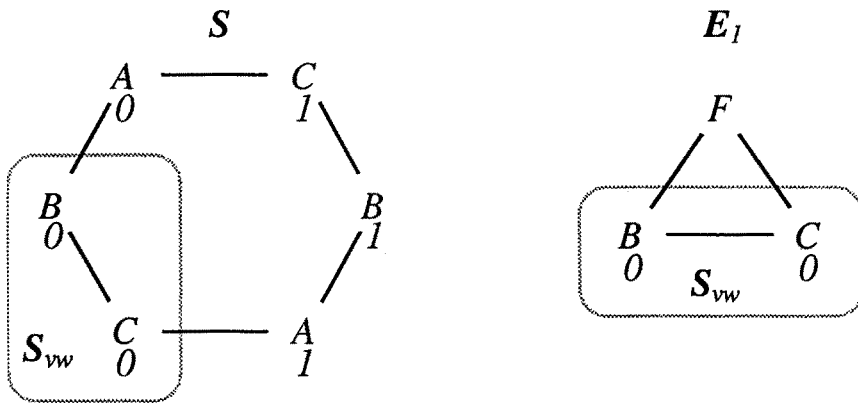
Now consider the third scenario, S_{xy} .



This scenario is the behavior of nodes x and y in S . It is also the behavior of nodes a and b in a correct behavior E_2 of G which results when they both run their devices on input 1, and node c is faulty, exhibiting the same behavior to node a that w exhibits to x in S , and the same behavior to node b that z exhibits to y in S . Validity requirements insure that nodes a and b must choose 1. Thus nodes x and y choose 1. But we have already established that node x must choose 0, a contradiction.

Now consider the general case of $|G| = n \leq 3f$. Partition the nodes of G into three sets, a , b and c , so that a , b and c have at least 1 and at most f nodes. This means that any two sets

together contain at least $n-f$ nodes. The nodes in each set are running agreement devices, and we denote by A the set of devices running at the nodes in a , and similarly for B and C . Now construct the covering graph S in the obvious way. Briefly, take two copies of G , and label the sets a, b and c in each copy by u, v and w , respectively, in one copy, and x, y and z in the other. Now replace the edges between nodes in u and w and between nodes in x and z by corresponding edges between u and z and between x and w . Assign devices to nodes of S according to their corresponding node in G . We represent the covering graph S and assigned devices exactly as above, so that the edges depicted between two sets of nodes in S , say sets u and v , are now a shorthand representation for all the edges in S between nodes in set u and nodes in set v . The inputs depicted for the sets of devices A, B and C are assigned to all the devices in the respective sets. The arguments proceed exactly as in the preceding pictures. We consider only one in detail.³



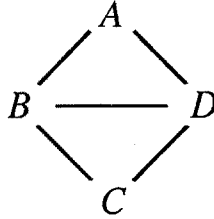
This scenario is now the behavior of the sets of nodes in v and w in the behavior S . It is the same as the behavior of the sets b and c in a behavior E_1 of G in which all nodes in both sets run their devices with input 0 and the nodes in set a exhibit the same behavior to members of b that the corresponding nodes in set u exhibit to the members of v in S , and the same behavior to nodes in c that the corresponding nodes in y exhibit to the members of x in S . Since sets b and c together contain at least $n-f$ correct nodes, E_1 is a correct behavior of G . Thus, all the nodes in b and c must decide 0, by the validity condition, and c contains at least one node, by construction.

³An alternative proof of the general bound may be obtained by a direct reductio to the case $f=1$. Given a system S and a partitioning of its communication graph G into subgraphs, there is a natural construction of a new system S' , obtained by collapsing the subgraphs into single nodes. The devices in S' are the (indexed) sets of devices running in each subgraph of G , the node behaviors of S' are the subsystem behaviors of the corresponding subgraphs in G , and the edge behaviors in S' are the corresponding sets of edge behaviors in S . Then the devices and behaviors in S' satisfy the Locality and Fault axioms if the underlying devices and behaviors in S do. Now, if Byzantine agreement were possible in a graph G with $|G| \leq 3f$ for some value of $f > 1$, this construction would imply that Byzantine agreement is possible for $f = 1$ in a subgraph of the triangle graph, contradicting Theorem 1. This is essentially the proof strategy for the general bound given in [PSL].

3.2. Connectivity

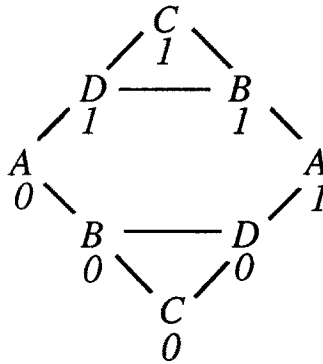
Now we carry out the $2f+1$ connectivity lower bound proof. Let $c(G)$ = connectivity of G . We assume we can achieve Byzantine agreement in a graph G with $c(G) \leq 2f$, and derive a contradiction.

For now, we consider the case $f=1$ and the communication graph G of four nodes a, b, c and d , running devices A, B, C and D , as indicated below.



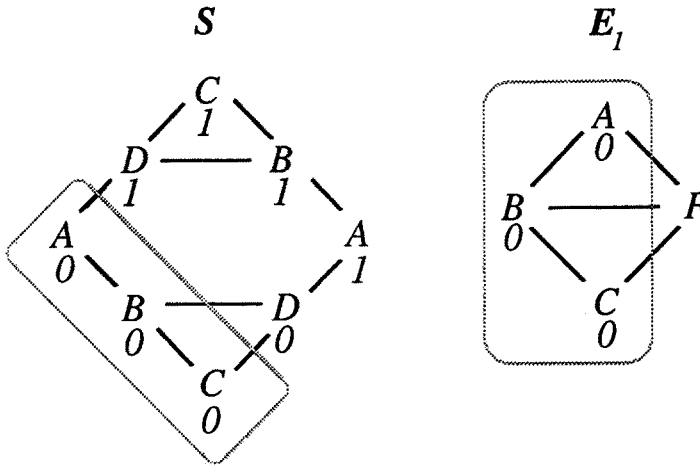
The connectivity of G is two; the two nodes b and d disconnect G into two pieces, the nodes a and c .

We consider the following system, with the eight-node graph S and devices and inputs as indicated.



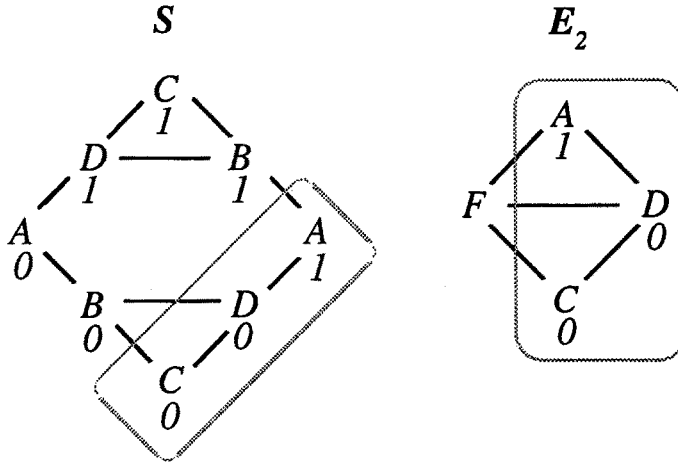
The resulting behavior of the system is S . We consider three scenarios in S : S_1, S_2 and S_3 .

The first scenario, S_1 , is shown below.



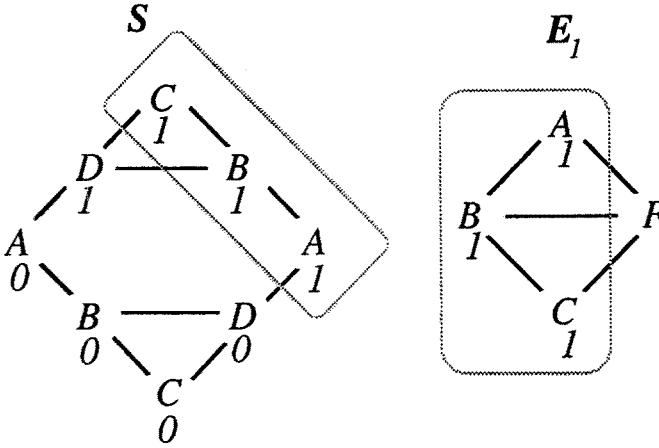
This is also a scenario in a correct behavior E_1 of G . In E_1 , nodes a, b and c are correct. Node d is faulty, exhibiting the same behavior to node a as one node running D in the covering graph, and the same behavior to b and c as the other node running D exhibits in the covering graph. Then nodes a, b and c must choose 0 in E_1 , and so must the nodes running A, B and C in S_1 .

Now consider the second scenario, S_2 .



This scenario in S is also a scenario in a correct behavior E_2 of G in which nodes c, d and a are correct. This time, node b is faulty, exhibiting the same behavior to nodes c and d as one node running B in the covering, and the same behavior to node a as the other node running B . So nodes a, c and d must agree in E_2 , and so do the corresponding nodes in S_2 . Since the node running C chooses 0 from the argument above, the nodes running D and A in S_2 choose 0 , too.

Finally, consider the last scenario S_3 .



This scenario is again the same as a scenario in a behavior E_3 of G in which nodes a , b and c are correct, but have input 1. Node d is faulty, exhibiting the same behavior to node a that one node running D in the covering graph exhibits, and the same behavior to nodes b and c as the other D in the covering exhibits. Then nodes a , b and c choose 1 in E_3 , and so must the nodes running A , B and C in S_3 , contradicting the argument above that the node running A chooses 0.

The general case for arbitrary $c(G) \leq 2f$ is an easy generalization of the case for $f = 1$. The same pictures are used. Just choose b and d to be sets consisting of at most f nodes each, such that removing the nodes in b and d from G disconnects two nodes u and v of G . Let G' be the graph obtained by removing b and d from G , let the set a contain those nodes connected to u , and the set c contain the remaining nodes of G' (c contains at least one node, v). Construct S as before, by taking two copies of G and rearranging edges between the ' a ' sets and their neighbors. The nodes and edges in our figures are now a shorthand for the actual nodes and edges of G and S .

This completes the proof of Theorem 1.

The succeeding impossibility results for other consensus problems follow the same general form as the two arguments above. We assume a problem can be solved by specific devices in an inadequate graph, G , install the devices in a graph S that covers G , and provide appropriate inputs. Using the Locality and Fault axioms, we argue the existence of a sequence of correct behaviors of G that have node and edge behaviors identical to some of those in the behavior of S . (This sequence was (E_1, E_2, E_3) , in the arguments above.) By the agreement condition, correct nodes in each of the behaviors of G have to agree. Because each successive pair of system behaviors has a correct node behavior in common, all of the correct nodes in all the behaviors in the sequence have to agree. But by the validity condition, correct nodes in the first behavior in the sequence must choose different values than those in the last behavior, a contradiction.

As we indicated in the introduction, a less general version of Theorem 1 was previously known, and the structure of our proof is very similar to that of earlier proofs [PSL], [D]. Our proof differs in the construction of the system behaviors E_1 , E_2 and E_3 . Earlier results construct these behaviors inductively using less general models of distributed systems. The detailed assumptions of the models are necessary to carry out the tedious and involved constructions.

Rather than construct the behaviors explicitly, we build them from pieces (node and edge behaviors) extracted from actual runs of the devices in a covering graph. The Locality and Fault axioms imply that scenarios in the covering graph are found in correct behaviors of the original inadequate graph.

The model used to obtain these results is an extremely general one, but it does assume that systems behave deterministically. (For every set of inputs, a system has a single behavior). By considering a system and inputs as determining a set of behaviors, nondeterminism may be introduced in a straightforward manner. One changes the Locality axiom to express the following. If there exist behaviors of two systems in which the inedge borders of two isomorphic subsystems are identical, there exist such behaviors in which the behaviors of the subsystems are also identical. Using this axiom, the same proofs suffice to show that nondeterministic algorithms cannot guarantee Byzantine agreement.

4. Weak Agreement

Now we give our impossibility results for the weak agreement problem. As in the Byzantine agreement case, nodes have Boolean inputs and must choose a Boolean output. The agreement condition is the same as for Byzantine agreement--all correct nodes must choose the same output. The validity condition, however, is weaker.

Agreement. Every correct node chooses the same value.

Validity. If all nodes are correct and have the same input, that input must be the value chosen.

The weaker validity condition has an interesting impact on the agreement problem. If any correct node observes disagreement or faulty behavior, then all are free to choose a default value, so long as they still agree.

Lamport notes that there are devices for reaching a form of approximate weak consensus which work when $|G| \leq 3f$. Running these for an infinite time produces exact consensus (at the limit) [L]. In such infinite behaviors any correct node observing disagreement or faulty behavior has plenty of time to notify the others before they choose a value. Thus, strengthening the choice condition by prohibiting such infinite solutions is necessary to obtain the lower bound.

If the communication delays are not bounded away from zero, a similar type of infinite behavior is possible. In fact, if there is no lower bound on transmission delay, and if devices can control the delay and have synchronized clocks, then we can construct an algorithm for reaching

weak consensus. This algorithm requires at most two broadcasts per node, each having non-zero transmission delay, and works with any number of faults. Again, this is because any correct node which observes disagreement or faulty behavior has plenty of time to notify the others before they choose a value.⁴ In more realistic models it is impossible to reach weak consensus in inadequate graphs. To show this, the minimal semantics introduced in the previous sections must be extended to exclude infinitary solutions. We do this as follows. Previously, behaviors of nodes and edges were elements of some arbitrary set. Henceforth, we consider them to be mappings from $[0, \infty)$ (our definition of time), to arbitrary state sets. Thus, if E_u is a behavior of node u , then u is in state $E_u(t)$ at time t .

We add the following condition to the weak agreement problem.

Choice. A correct node must choose 0 or 1 after a finite amount of time.

This means there is a function CHOOSE from behaviors of nodes running weak agreement devices to $\{0,1\}$, with the following property: Every such behavior E has a finite prefix E^t (E restricted to the interval $[0,t]$) such that all behaviors E' extending E^t have $\text{CHOOSE}(E) = \text{CHOOSE}(E')$.

This choice condition prohibits Lamport's infinite solution. To prohibit the second solution, we bound the rate at which information can traverse the network. To do so, we add the following stronger locality axiom to our model.

Bounded-Delay Locality Axiom.

There exists a positive constant δ such that the following is true. Let G and G' be systems with behaviors E and E' , respectively, and isomorphic subsystems U and U' , (with vertex sets U and U'). If the corresponding behaviors of the inedge borders of U and U' in E and E' are identical through time t , then scenarios E_U and $E_{U'}$ are identical through time $t+\delta$.

Thus, news of events k edges away from some subgraph G' takes time at least $k\delta$ to arrive at G' . In a model with explicit messages, this axiom would hold if the transmission delay is at least δ ; the edge behaviors in our model would correspond to state descriptions of the transmitting end of each communications link.

Theorem 2: *Weak agreement is not possible in inadequate graphs for models satisfying the Bounded-Delay Locality axiom.*

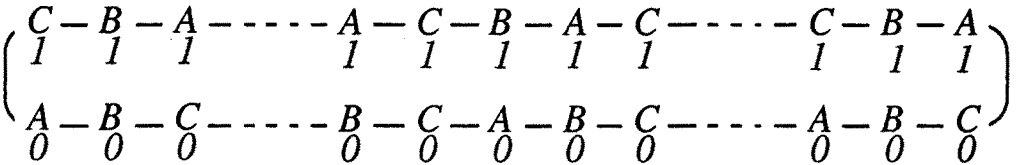
Again, we first sketch the $3f+1$ node bound. In this case the previously published proof [L]

⁴Nodes start at time 0, and decide at time 1. They broadcast their value at time 0, specifying it to arrive at time 1/2. If a node first detects disagreement or failure (at time 1-t), it broadcasts a "failure detected, choose default value" message, specifying it to arrive at time 1-t/2. The obvious decision is made by everyone at time 1.

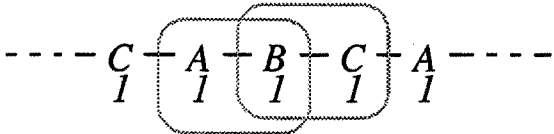
was very difficult. As before, we restrict our attention to the case $|G| = n = 3, f = 1$. (The case for general f follows immediately, just as above.)

Assume there are weak agreement devices A, B and C , for the triangle graph G containing nodes a, b and c . Consider the two behaviors of G in which all nodes are correct, and all have input 0 or all have input 1. Let t' be an upper bound on the time it takes all nodes to choose 0 or 1 in both behaviors. Choose $k \geq t'/\delta$ to be a multiple of 3.

The covering graph S consists of $4k$ nodes, arranged in a ring and assigned devices and inputs as follows:

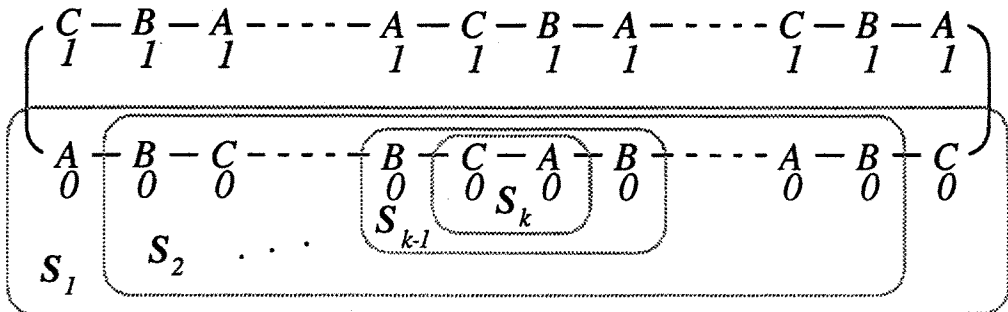


Consider the resulting behavior S , and each pair of successive two-node scenarios, such as the two below.



As before, each scenario is identical to a scenario in a behavior in G of the appropriate two weak consensus devices. Since each pair of successive scenarios overlaps in one node behavior (here, that of the node running B), the agreement condition requires that all the nodes in both scenarios must choose the same value in G and in S . Thus, every node in S must choose the same value. Without loss of generality, assume they choose 1.

Consider the k scenarios indicated below.



Let E be the behavior of G in which a, b and c are correct and each has input 0, and denote the resulting behaviors of a, b and c by E_a, E_b and E_c , respectively.

Lemma 3: *The behavior in scenario S_i of a node running device A (or B or C) is identical to E_a (or E_b or E_c) through time $i\delta$.*

Proof: An easy induction using the Bounded-Delay Locality axiom, essentially arguing that no device in S_i can hear from a device with input 1 until after time $i\delta$.

By Lemma 3, the nodes running devices C and A in scenario S_k have behaviors identical to E_c and E_a through time $k\delta \geq t'$. Since nodes c and a in G have chosen output 0 by this time, so have the corresponding nodes in S_k , a contradiction.

The general case of $|G| \leq 3f$ and the connectivity bound follow as for Byzantine agreement.

There are strong similarities between this argument and a proof by Angluin, concerning leader elections in rings and arbitrarily long lines of processors [A]. Both results depend crucially on the existence of a lower bound on the rate of information flow. Under this assumption, devices in different communication networks can be shown to observe the same local behavior for some fixed time.

5. Byzantine Firing Squad

The Byzantine firing squad problem addresses a form of synchronization in the presence of Byzantine failures. The problem is: given an input stimulus, synchronize the response of entering a designated FIRE state. This problem was studied originally in [BL]. In [CDDS], a reduction of weak agreement to the Byzantine firing squad problem demonstrates that the latter is impossible to solve in inadequate graphs. We provide a direct proof that a simple variant of the original problem is impossible to solve in inadequate graphs. (In the original version, the stimulus can arrive at any time. We require that it arrive either at time 0, or not at all. Our validity condition is slightly different.) The proof is very similar to that for weak agreement.

One or more devices may receive a stimulus at time 0. We model the stimulus as an input of 1, and the absence of the stimulus as an input of 0. Correct executions must satisfy the following conditions.

Agreement. If a correct node enters the FIRE state at time t , every correct node enters the FIRE state at time t .

Validity. If all nodes are correct and the stimulus occurs at any node, all nodes enter the FIRE state after some finite delay. If the stimulus does not occur and all nodes are correct, no node ever enters the FIRE state.

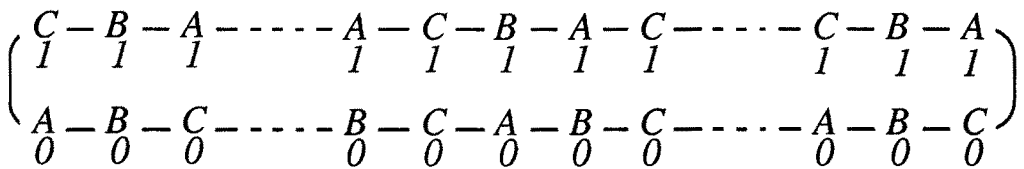
As in the case of weak agreement, solutions to the Byzantine firing squad problem exist in models in which there is no minimum communication delay. Thus, the following result requires the Bounded-Delay Locality axiom, in addition to the Fault axiom.

Theorem 4: *The Byzantine firing squad problem cannot be solved in inadequate graphs for models satisfying the Bounded-Delay Locality axiom.*

We sketch the $3f+1$ node bound. As before, we examine the case $|G| = n = 3, f = 1$.

Assume there are Byzantine firing squad devices A, B and C for the triangle graph G containing nodes a, b and c . Consider the two behaviors of G in which all nodes are correct, and all have input 0 or all have input 1. Let t be the time at which the correct devices enter the FIRE state in the case that the stimulus occurred (the input 1 case). Choose $k \geq t/\delta$ to be a multiple of 3. (Recall that δ is the minimum transmission delay defined in the Bounded-Delay Locality axiom).

The covering graph S consists of $4k$ nodes, arranged in a ring and assigned devices and inputs as follows:



Similarly to the proof for weak agreement, the middle two devices receiving the stimulus enter the FIRE state at time t , as their behavior through time t is the same as that of the correct nodes in G which have received the stimulus and fire at time t . Because of the communication delay, there is not enough time for "news" from the distant nodes to reach these devices. By repeated use of the agreement property, all the devices in S must fire at time t . But through time t , the middle two devices not receiving the stimulus behave exactly as correct nodes in G which do not receive the stimulus (the input 0 case). Thus, they do not fire at time t , a contradiction.

6. Approximate Agreement

Next, we turn to two versions of the approximate agreement problem [DLPSW,MS]. We call them *simple approximate agreement* and $(\epsilon, \delta, \gamma)$ -*agreement*. In these problems nodes have real values as inputs and choose real numbers as a result. The goal is to have the results close to each other and to the inputs. To obtain the strongest possible impossibility result, we formulate very weak versions of the problems.

For the following two theorems we use only the Locality and Fault axioms. We do not need the Bounded-Delay Locality axiom used for the weak agreement and firing squad results.

6.1. Simple Approximate Agreement

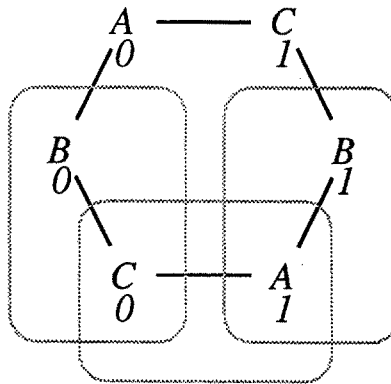
First, examine a version of the simple approximate agreement problem [DLPSW]. Each correct node has a real value from the interval $[0,1]$ as input, runs its device, and chooses a real value. Correct behaviors (those in which at least $n - f$ nodes are correct) must satisfy the following conditions.

Agreement. The maximum difference between values chosen by correct nodes must be strictly smaller than the maximum difference between the inputs, or be equal to the latter difference if it is zero.

Validity. Each correct node chooses a value within the range of the inputs of the nodes.

Theorem 5: *Simple approximate agreement is not possible in inadequate graphs.*

The proof is almost exactly that for Byzantine agreement. Here, we consider devices which take as inputs numbers from the interval $[0,1]$, and choose a value from $[0,1]$ to output. (Outputs are modeled by a function CHOOSE from behaviors of nodes running the devices to the interval $[0,1]$.) As before, assume simple approximate agreement can be reached in the triangle graph G . Consider the following three scenarios from the indicated behavior in the covering graph S .



Again, each scenario is also a scenario in a correct behavior of G . In the first scenario, the only value C can choose is 0. In the third, the only value A can choose is 1. This means the values chosen by A and C in the second scenario are 0 and 1, so that the outputs are no closer than the inputs, violating the agreement condition.

The general case of $|G| \leq 3f$ and the connectivity bounds follow as for Byzantine agreement.

6.2. $(\epsilon, \delta, \gamma)$ -Agreement

This version of approximate agreement is based on that in [MS]. Let ϵ , δ and γ be positive real numbers. The correct nodes receive real numbers as inputs, with r_{min} and r_{max} the smallest and largest such inputs, respectively. These inputs are all at most δ apart (i.e. the interval of inputs $[r_{min}, r_{max}]$ has length at most δ). They must choose a real number as output, such that correct behaviors (those in which at least $n - f$ nodes are correct) satisfy the following conditions.

Agreement. The values chosen by correct nodes are all at most ϵ apart.

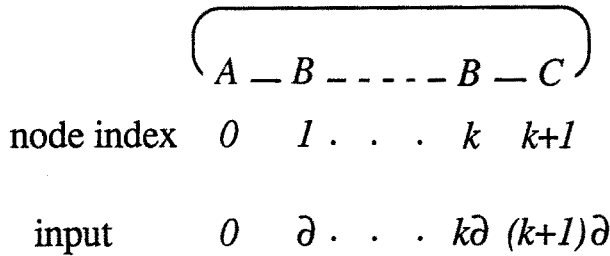
Validity. Each correct node chooses a value in the interval $[r_{min} - \gamma, r_{max} + \gamma]$.

Note that if $\epsilon \geq \delta$, $(\epsilon, \delta, \gamma)$ -agreement can be achieved trivially by choosing the input value as output.

Theorem 6: *If $\epsilon < \delta$, $(\epsilon, \delta, \gamma)$ -agreement is not possible in inadequate graphs.*

Proof: Let ϵ , δ and γ be positive real numbers with $\epsilon < \delta$. We prove only the $3f+1$ bound on the number of nodes. Assume that devices A, B and C exist which solve the $(\epsilon, \delta, \gamma)$ -approximate agreement problem in the complete graph G on three nodes for particular values of ϵ , δ and γ , where $\epsilon < \delta$.

Choose k sufficiently large that $\delta > 2\gamma/(k-1) + \epsilon$, and $k+2$ is divisible by three. The covering graph S contains $k+2$ nodes arranged in a ring, with devices and inputs assigned to create the following system.



Let S_i , for $0 \leq i \leq k$, denote the two-node scenario in S containing the behaviors of nodes i and $i+1$. By the Fault Axiom, each scenario S_i is a scenario of a correct behavior of G , in which the largest input value to a correct node is $(i+1)\delta$.

Lemma 7: *For $0 \leq i \leq k$, the value chosen by the device at node $i+1$ is at most $\delta + \gamma + i\epsilon$.*

Proof: The proof is a simple induction. By validity applied to scenario S_0 , the device at node 1 chooses at most $\delta + \gamma$. Assume inductively that the device at node i chooses at most $\delta + \gamma + (i-1)\epsilon$, for $0 < i < k+1$. By agreement applied to scenario S_i , the device at node $i+1$ chooses at most $\delta + \gamma + i\epsilon$.

In particular, Lemma 7 implies the device at node k chooses at most $\delta + \gamma + (k-1)\epsilon$. But validity applied to scenario S_k implies the device at node k chooses at least $k\delta - \gamma$. So $k\delta - \gamma \leq \delta + \gamma + (k-1)\epsilon$. This implies $\delta \leq 2\gamma/(k-1) + \epsilon$, a contradiction.

The general case of $|G| \leq 3f$ and the connectivity bounds follow as in previous proofs.

7. Clock Synchronization

Each node has a hardware clock and maintains a logical clock. The hardware clocks are real-valued, invertible and increasing functions of time. Since different hardware clocks run at different rates, it may be necessary to synchronize the logical clocks more closely than the

hardware clocks. In addition, logical clocks should be reasonably close to real time--setting them to be constantly zero, for example, should be forbidden. Thus, we require the logical clocks to stay within some envelope of the hardware clocks. [See the paper by Lundelius, Lynch and Simons in this volume.]

This problem was studied in [DHS] for the case of linear clock and envelope functions, where it was shown that it is impossible to synchronize to within a constant in inadequate graphs. Some more general synchronization issues were raised, such as that diverging linear clocks can be synchronized to within a constant if nodes can run their logical clocks as the logarithm of their hardware clocks. For a large class of clock and envelope functions (increasing and invertible clocks, non-decreasing envelopes), we can characterize the best synchronization possible in inadequate graphs. This synchronization requires no communication whatsoever.

We model node i 's hardware clock, D_i , as an input to the device at node i that has value $D_i(t)$ at time t . The value of the hardware clock at time t is assumed to be part of the state of the node at time t . The time on node i 's logical clock at real time t is given by a function of the entire state of node i . Thus, if E_i is a behavior of node i (such that node i is in state $E_i(t)$ at time t), then we express i 's logical clock value at time t as $C_i(E_i(t))$.

We assume that any aspect of the system which is dependent upon time (such as transmission delay, minimum step time, maximum rate of message transmission) is a function of the states of the hardware clocks. Having made this assumption, it is clear that speeding up or slowing down the hardware clocks uniformly in a behavior E cannot be observable to the nodes, so that the only impact on a E should be to change the (unobserved) real times at which events occur.

To formalize this assumption, we need to talk about *scaling* clocks and behaviors. Let h be any invertible function of time. If E is a behavior (of a edge or node), then Eh , the behavior E scaled by h , is such that $Eh(t)=E(h(t))$, for all times t . Similarly, Dh is the hardware clock D scaled by h : $Dh(t)=D(h(t))$. If E is a system behavior or scenario, Eh is the system behavior or scenario obtained by scaling every node and edge behavior in E by h . Similarly, if S is a system, then Sh is the system obtained by scaling every clock in S by h . Intuitively, a scaled clock or behavior is in the state at time t that the corresponding unscaled clock or behavior is in at time $h(t)$.

Scaling Axiom If E is the behavior of system S , then Eh is the behavior of system Sh .

If this axiom is significantly weakened, as by bounding the transmission delay, clock synchronization may be possible in inadequate graphs.

In the following we use the Locality, Fault and Scaling axioms. We do not need the Bounded-Delay Locality axiom used for the weak agreement and firing squad results.

The synchronization problem can be stated as follows. Let correct hardware clocks run either

at $p(t)$ or $q(t)$, where p and q are increasing, invertible functions, with $p(t) \leq q(t)$, for all t . Let the envelope functions l and u be non-decreasing functions such that $l(t) \leq u(t)$, for all t .

Consider what happens if all logical clocks are run at the lower envelope, $C(E(t))=l(D(t))$. Then the logical clocks are synchronized to within $l(q(t))-l(p(t))$. The goal then, is to improve this trivial synchronization. We show that logical clocks cannot be synchronized to within $l(q(t))-l(p(t))-\alpha$, for any positive α . That is, *nontrivial synchronization* is achieved by synchronization devices in G if there exist positive constant α and time t' such that every correct system behavior E satisfies the following conditions.

Agreement. For any two correct nodes i and j in E , $|C_i(E_i(t)) - C_j(E_j(t))| \leq l(q(t)) - l(p(t)) - \alpha$, for all times $t \geq t'$.

Validity. For any correct node i in E , with hardware clock D_i and resulting behavior E_i , $l(p(t)) \leq C_i(E_i(t)) \leq u(q(t))$.

Theorem 8: *Nontrivial synchronization is not possible in inadequate graphs for models satisfying the Scaling axiom.*

The argument is very similar to the proof of Theorem 6. We construct a covering graph and assign hardware clocks in such a way that each node seems to be running slow relative to one neighbor, and fast relative to the other. As usual, the Fault axiom allows us to draw conclusions about behaviors in the covering system from required behavior in the original graph. The Agreement condition requires the nodes to be synchronized with their faster neighbor. An induction shows that the slowest node (whose fast neighbor is synchronized with a faster neighbor, in turn synchronized with an even faster neighbor...) must run so fast as to violate the upper envelope condition.

Specifically, we show that for every integer $k > 2$, there is a behavior E of G in which node i is correct, has hardware clock $D_i = p$ (that is, $D_i(t) = p(t)$), and in which $C_i(E_i(t')) \geq l(p(t')) + k\alpha$. For k big enough, this violates the upper envelope condition, $C_i(E_i(t')) \leq u(q(t'))$.

Define $h = p^{-1}q$. (That is, $h(t) = p^{-1}(q(t))$.) Then $h^{-1} = q^{-1}p$. Note that $h(t) \geq t$ for all t , since $p(t) \leq q(t)$.

We begin with the three node, one fault case.

Assume the existence of devices A , B and C , time t' and positive constant α such that logical clocks of correct nodes obey the agreement and validity conditions:

$$|C_i(E_i(t)) - C_j(E_j(t))| \leq l(q(t)) - l(p(t)) - \alpha, \text{ for all times } t \geq t'.$$

$$l(p(t)) \leq C(E_i(t)) \leq u(q(t)), \text{ for all times } t.$$

Choose an integer $k > 2$, such that $k+2$ is a multiple of three, and such that $l(p(t')) + k\alpha >$

$u(q(t))$. The covering graph S contains $k+2$ nodes arranged in a ring, with devices and clock inputs assigned to create the following system.

	$\overbrace{A \text{ --- } B \text{ - - - - } B \text{ --- } C}$					
node index	0	1	\dots	k	$k+1$	
hardware clock	g	gh^{-1}	\dots	gh^{-k}	$gh^{-(k+1)}$	
behavior	E_0	E_1	\dots	E_k	E_{k+1}	

Let S be the behavior of this system. An initially troubling concern is that the hardware clocks in S are much slower in most of the devices in S than they would be in a correct behavior in G . But consider S_i , the two-node scenario containing the behaviors of nodes i and $i+1$, where $0 \leq i \leq k$.

	$\text{---} A \text{ --- } B \text{ ---}$	
node index	i	$i+1$
hardware clock	ph^{-i}	$ph^{-(i+1)}$
behavior	E_i	E_{i+1}

Now consider $S_i h^i$, the scenario S_i scaled by h^i .

	$\text{---} A \text{ --- } B \text{ ---}$	
node index	i	$i+1$
hardware clock	p	q
behavior	$E_i h^i$	$E_{i+1} h^i$

In this scenario, the hardware clocks have values within the constraints for correct behaviors of G . Thus we have the following.

Lemma 9: *Scenario $S_i h^i$, for $0 \leq i \leq k$, is a scenario containing the behaviors of two correct nodes in a correct behavior of G .*

Lemma 10: For all i , $0 \leq i \leq k$, and all $t \geq h^i(t')$, $|C_{i+1}(E_{i+1}(t)) - C_i(E_i(t))| \leq l(q(h^i(t))) - l(p(h^i(t))) - \alpha$.

Proof: Fix $t \geq h^i(t')$. Then $h^i(t) \geq t'$. By Lemma 9, i and $i+1$ are correct in $S_i h^i$, so by the agreement assumption $|C_{i+1}(E_{i+1} h^i(h^i(t))) - C_i(E_i h^i(h^i(t)))| \leq l(q(h^i(t))) - l(p(h^i(t))) - \alpha$. The result is immediate.

Let time $t'' = h^k(t')$. Note that $t'' \geq h^i(t')$, for $i \leq k$.

Lemma 11: For all i , $1 \leq i \leq k+1$, $C_i(E_i(t'')) \geq l(qh^i(t'')) + (i-1)\alpha$.

Proof: The proof is by induction on i . By Lemma 9, scenario S_0 is a scenario in G of correct nodes a and b with hardware clocks q and p , respectively. From the validity condition, for all t , $C_1(E_1(t)) \geq l(p(t))$. Setting $t = t''$, and substituting qh^{-1} for p , we have the basis step: $C_1(E_1(t'')) \geq l(qh^{-1}(t''))$.

Now make the inductive assumption $C_i(E_i(t'')) \geq l(qh^i(t'')) + (i-1)\alpha$, for $1 \leq i \leq k$.

Since $t'' \geq h^i(t')$, from Lemma 10, we know $|C_{i+1}(E_{i+1}(t'')) - C_i(E_i(t''))| \leq l(qh^i(t'')) - l(ph^i(t'')) - \alpha$.

This implies $C_{i+1}(E_{i+1}(t'')) \geq C_i(E_i(t'')) - l(qh^i(t'')) + l(ph^i(t'')) + \alpha$.

Substituting for $C_i(E_i(t''))$ using the inductive assumption gives us $C_{i+1}(E_{i+1}(t'')) \geq l(qh^i(t'')) - l(qh^i(t'')) + l(ph^i(t'')) + i\alpha = l(ph^i(t'')) + i\alpha$. Noting that $p = qh^{-1}$, we have the result, $C_{i+1}(E_{i+1}(t'')) \geq l(qh^{-(i+1)}(t'')) + i\alpha$.

Proof of Theorem 8:

Lemma 11 implies $C_{k+1}(E_{k+1}(t'')) \geq l(qh^{-(k+1)}(t'')) + k\alpha$. Since $t'' = h^k(t')$, we have $C_{k+1}(E_{k+1}(t'')) = C_{k+1}(E_{k+1}(h^k(t'))) = C_{k+1}(E_{k+1} h^k(t')) \geq l(qh^{-(k+1)} h^k(t')) + k\alpha = l(p(t')) + k\alpha$.

But the upper envelope constraint for the scaled scenario $S_k h^k$ (in which $k+1$ is correct and has hardware clock $p(t)$) implies that $C_{k+1}(E_{k+1} h^k(t')) \leq u(q(t'))$. Thus, $l(p(t')) + k\alpha \leq u(q(t'))$. This violates the assumed bound on k , $l(p(t')) + k\alpha > u(q(t'))$.

Once again, the general case of $|G| \leq 3f$ is a simple extension of this argument. The connectivity bound also follows easily, as with the earlier results.

7.1. Linear Envelope Synchronization and other Corollaries

Linear envelope synchronization, as defined in [DHS], examines the synchronization problem when the clocks and envelope functions are linear functions ($q(t)=rt$, $p(t)=t$, $l(t)=at+b$ and $u(t)=ct+d$). It requires correct logical clocks to remain within a constant of each other, so that the agreement condition is $|C_i(E_i(t)) - C_j(E_j(t))| \leq \alpha$, for all times t , instead of our weaker

condition $|C_i(E_i(t)) - C_j(E_j(t))| \leq art - at - \alpha$, for all times $t \geq t'$. Our validity condition is slightly weaker, as well. Thus, the proof of [DHS] shows that logical clocks cannot be synchronized to within a constant; we show that the synchronization of logical clocks cannot be *improved* by a constant over the synchronization ($art - at$) that can be achieved trivially. Thus the following corollary follows immediately from Theorem 8. (Each of the four corollaries below holds for models satisfying the Scaling axiom.)

Corollary 12: *Linear envelope synchronization is not possible in inadequate graphs.*

By choosing specific values for the clock and lower envelope functions, we get the following additional results immediately from Theorem 8. Note that the particular choice of the upper envelope function does not affect the minimal synchronization possible in inadequate graphs, although the existence of *some* upper envelope function is necessary to obtain our impossibility proofs.

Corollary 13: *If $p(t)=t$, $q(t)=rt$, and $l(t)=at+b$, no devices can synchronize a constant closer than $art-at$ in inadequate graphs.*

Corollary 14: *If $p(t)=t$, $q(t)=t+c$ and $l(t)=at+b$, no devices can synchronize a constant closer than ac in inadequate graphs.*

Corollary 15: *If $p(t)=t$, $q(t)=rt$ and $l(t)=\log_2(t)$, no devices can synchronize a constant closer than $\log_2(r)$ in inadequate graphs.*

In general, the best possible synchronization in inadequate graphs can be achieved without any communication at all. The best that nodes can do is run their logical clocks as slowly as they are permitted, $C(E(t)) = l(D(t))$.

8. Conclusion

Most of the results we have presented were previously known. While our proofs are both simpler than earlier proofs, and apply to more general models, these are not the main contributions. The simplicity and generality are welcome byproducts of our attempt to identify the fundamental issues and assumptions behind a collection of similar results.

One important contribution is to elucidate the relationship between the unrestricted or Byzantine failure assumption and inadequate graphs. As is clear from our proofs, this fault assumption permits faulty nodes to mimic executions of disparate network topologies. If the network is inadequate, a covering graph can be constructed so that correct devices cannot distinguish the execution in the original graph from one in the covering graph.

A second contribution is related to the generality of our results. Nowhere do we restrict state sets or transitions to be finite, or even to reflect the outcome of effective computations. The

inability to solve consensus problems in inadequate graphs has nothing to do with computation *per se*, but rather with distribution. It is the distinction between local and global state, and the uncertainty introduced by the presence of Byzantine faults, which result in this limitation.

Finally, we have identified a small, natural set of assumptions upon which the impossibility results depend. For example, in the case of weak agreement and the firing squad problem, the correctness conditions are sensitive to the actions of faulty nodes. Instantaneous notification of the detection of fault events would allow one to solve these problems. An assumption that there are minimum delays in discovering and relaying information about faults is sufficient to make these problems unsolvable.

9. References

- [A] D. Angluin, "Local and Global Properties in Networks of Processors," *Proc. of the 12th STOC*, April 30-May 2, 1980, Los Angeles, CA., pp. 82-93.
- [B] J. Burns, "A Formal Model for Message Passing Systems," TR-91, Indiana University, September 1980.
- [BL] J. Burns, N. Lynch "The Byzantine Firing Squad Problem," submitted for publication.
- [CDDS] B. Coan, D. Dolev, C. Dwork and L. Stockmeyer "The Distributed Firing Squad Problem," *Proc. of the 17th STOC*, May 6-8, 1985, Providence R.I.
- [D] D. Dolev, "The Byzantine Generals Strike Again," *Journal of Algorithms*, 3, 1982, pp. 14-30.
- [DHS] D. Dolev, J. Halpern, H. Strong, "On the Possibility and Impossibility of Achieving Clock Synchronization," *Proc. of the 16th STOC*, April 30-May 2, 1984, Washington, D.C., pp. 504-510.
- [DLPSW] D. Dolev, N. A. Lynch, S. Pinter, E. Stark and W. Weihl, "Reaching Approximate Agreement in the Presence of Faults," *Proc. of the 3rd Annual IEEE Symp. on Distributed Software and Databases*, 1983.
- [IR] A. Itai, M. Rodeh, "The Lord of the Ring or Probabilistic Methods for Breaking Symmetry in Distributive Networks," RJ-3110, IBM Research Report, April 1981.
- [L] L. Lamport, "The Weak Byzantine Generals Problem", *JACM*, 30, 1983, pp. 668-676.
- [LSP] L. Lamport, R. Shostak, M. Pease, "The Byzantine Generals Problem," *ACM Trans. on Programming Lang. and Systems* 4, 3 (July 1982), 382-401.
- [MS] S. Mahaney, F. Schneider, "Inexact Agreement: Accuracy, Precision, and Graceful Degradation," *Proc. of the 4th Annual ACM Symposium on Principles of Distributed Computing*, August 5-7, 1985, Minacki, Ontario.
- [PSL] M. Pease, R. Shostak, L. Lamport, "Reaching Agreement in the Presence of Faults," *JACM* 27:2 1980, 228-234.