Simulating Population Protocols

Mike Ruberry
Yale University
312 404 1595
michael.ruberry@yale.edu

ABSTRACT
We create a simulator and implement four protocols described for the population protocol model created by Professors Aspnes, Angluin, Eisenstat, and Fischer. The protocols are examined for run characteristics and analyzed for alterations which may govern their performance. We conjecture that the one-way epidemic protocol closely fits a bound of $1.5 \times n \log n$, and that the phase count protocol is reasonably considered $\frac{3}{4} \times n \log n$. Furthermore, we proffer evidence which suggests that the potential for small cuts between the infected and uninfected set of nodes in the one-way epidemic protocol governs its behavior more than connectedness of the graph, demonstrating that even with scantily few edges the protocol performs as if the graph were completely connected. We also suggest that certain protocols in this model are too prone to waiting for a few nodes to converge, and they should be reconsidered. Either new protocols need to be made with the same effect, or a method in which only near-total convergence is required developed. This would greatly enhance the model’s potential for real-world operation. Finally, we present evidence that it is not the size of the problem as much as the constraints it imposes which determine run-time of these protocols. This implies that sensor networks which may use this model of computation should not naively seek maximum connectivity, but must accept a trade-off for functional operation.

INTRODUCTION
The population protocol model described in [1], [2], and [3] is one in which a fair but possibly adversarial scheduler picks two nodes who share an edge in a graph to interact at each round. The nodes have limited memory, they are deterministic, they often are incapable of naming themselves or determining if they are interacting with a node they have seen before, or one which is completely new. Additionally, their interaction is governed entirely by a predefined transition function. For more exacting formal detail please see [3].

While interesting theoretical results abound for this model, little is known about the actual running of the various theoretic protocols. Their time to convergence, the state in which no interaction will cause a node to alter its state, for example, is usually unknown beyond a theoretic bound. That is often good enough, yet the phase clock protocol discussed here requires knowledge of the constants to function properly.

Additionally, theory does not always catch every detail. It can permit us to function in an abstract fashion instead of knowing a model intimately. Simulation demands we do the latter.

In this paper we simulate four major protocols on the population protocol model. We discuss their run time and the conditions which alter it. We provide reasonable constants for two protocols, one-way epidemic and phase clock, both of which required that knowledge if they were to be implemented. We present data which suggests that pragmatically effective protocols in this model may have to sacrifice convergence, and that certain structural attributes of graphs, such as their edge density, are more important than their degree of connectivity.

METHODOLOGY
The trials performed in this paper are results from a simulator built to emulate the population protocol model. As in the construction of any artifact from a theoretical model there were some changes and alterations. Protocols which were under specified had assumptions made on their behalf, and the scheduler became random instead of adversarial. We do not believe either of these acts had a noticeable effect on the functioning of the protocols presented here. In particular, assumptions made regarding the protocols are mostly confined to initial values for state variables. These assumptions are made clear in the protocols’ subsections.

The high degree of variation or deviation from the mean in these trials is endemic to the simulation of this model. This is unfortunate, but there is nothing to be done for it except to run more trials. Due to the limitations of the simulator usually only 100 trials were run per data point. While we believe this to be sufficient others may differ. The number of nodes used to form the graphs some protocols operated on can be painfully small, sometimes as low as 100 or 200 nodes. While there may be effects going unnoticed because the graphs are not sufficiently complex
this seems unlikely. The level of scrutiny applied here is not sufficient to detect the minor variations or patterns that may be found in much larger graphs when using complex protocols. The simpler protocols covered in this paper do make use of large numbers of nodes.

SIMULATIONS

The following section details the various trials performed using the simulator. It should be noted that detailed theoretical analysis of all of these protocols are present in the references. I have attempted to add pragmatic data where applicable, such as the type of graphs these protocols require and their starting states.

It should be noted that through this paper the term ‘round’ is used in place of ‘interaction.’ The notable exception to this is during the discussion of the phase clock protocol, in which ‘round’ refers to a constant number of phases. During that section the term ‘interaction’ will be used in place of ‘round,’ except on the graphs, which maintain the nomenclature of the other sections.

Epidemics

We simulated the one-way epidemic protocol discussed in [1]. This protocol is simple, yet it forms the basis for more advanced protocols and is related to the next protocol discussed. Here, ‘one-way’ is used to mean that only the responder’s state is altered in an interaction. For the epidemic protocol to converge with probability = 1 it requires a strongly connected graph. As a result we will restrict our inquiry to undirected connected graphs.

The protocol has only two states: 0 or 1, and a single transition rule: \((Q_{init}, Q_{resp}) \rightarrow (Q_{init}, max(Q_{init}, Q_{resp}))\). In English, if the initiator’s state is 1, or ‘infected,’ then the responder’s state is set to 1 as a result of the interaction. If the initiator is not infected there is no change in the state of either node. The protocol begins with one node infected, all other nodes begin uninfected.

This protocol is \(\Theta(n \log n)\) with high probability[1], a bound that we found held remarkably well. Figure 1 shows data collected from several simulated trial runs of the protocol on complete graphs. The data points for convergence over all trials are in blue. The pink line is the function \(y = 1.5 \times \log x\), where \(x = n\). The tan triangles are placed at the average of all trials for each node value used.

The clear correspondence between this simple function and the average of values for various values of \(n\) is surprising. The utility of this constant will become clear in the discussion of the phase clock protocol, but the low value informs the utility of epidemic-like protocols, since they are likely to converge quickly.

The high deviation of the number of rounds required for completion is likely due to the difficulty with infecting the last few nodes in a graph. As discussed in [1] there is high variance associated with these nodes. Figure 2 charts the course of several runs of the epidemic protocol. Note the large number of rounds required for the last few nodes to become infected.

Also of interest in Figure 2 is the shape of the graph. After an initial start-up period in which the infection spreads slowly it suddenly and quickly expands to cover most of the graph. A protocol which only requires a high probability of nodes to be in the correct state could benefit greatly from this fact, since nearly half of the time required for convergence is used to infect the last 20% of nodes.

The similarity between the start and end of the protocol should not be understated. At the beginning of the protocol few edges will spread the infection, just as the case is at the end. In the middle, however, there is a large set of edges which form the cut between the infected and uninfected node sets. The size of the cut between these sets will be an important notion in understanding the behavior of many population protocols, particularly the ones presented here. To better understand its importance, consider if the scheduler selects an edge which is not an element of the cut between the infected an uninfected nodes. This edge is incident on two infected or two uninfected nodes. Either way, the epidemic makes no progress. Only edges over the cut advance the protocol. If at any point in the protocol there is a small cut the protocol will likely wait a long time before an edge in that cut is chosen. For an extreme example of this see the cancellation protocol below.

This behavior is present even if we do not deal with complete graphs. Figure 3 charts trial runs on random graphs with a degree bound of 3. To ensure connectivity the graphs were generated with 1000 nodes and the largest connected component was used for the trials—hence the slight variance in the number of infected nodes between trials. The average number of nodes in the largest connected component was slightly more than 900 under these conditions.

While this general behavior is to be expected based on our previous analysis, the longer start-up and finishing times are an oddity. The slopes through the ‘middle portions’ of Figures 2 and 3 are close, so the relative size of the cut is roughly the same in both instances. It takes more nodes in graphs with low degree to achieve the same cut, however. A possible explanation for this is that the random distribution of edges causes subgraphs with a high ratio of edges to become infected quickly, but then languish due to a relative lack of nodes over their cut with the uninfected nodes. These scenarios in which small cuts may form are bottlenecks and need to be avoided. Uniform density of edges throughout the graph prevents this from happening.

Figure 4 adds credence to this theory. For graphs with sufficient random connectivity, represented here by the
crude metric of the number of edges present in the random graph, the protocol’s behavior is nearly indistinguishable from that of the completely connected case. These graphs, being randomly distributed, are likely to have the property that any subgraph looks much like any other. That is, the graphs have the property that no particular subgraph is denser (has a higher edge to node ratio) than any other. However, once connectivity dips too low there is a dramatic increase in the number of rounds required for the protocol to converge, as well as the range of values. This wider range of rounds required for convergence suggests that a small set of edges is often important in graphs with low connectivity, as the density of edges is subject to greater variation. In these cases there is a small cut between infected and uninfected nodes.

Figure 5 further demonstrates that beyond some threshold of edges the protocol’s operation is close to that on a complete graph. Here, with only 2% connectivity, or an average of 20 edges per node instead of 999, there is only a slight increase in the number of rounds required for convergence. Clearly the total number of connections is not relevant to the proper functioning of the protocol, but rather their distribution over the graph. For fast convergence uniformity of density of edges among the subgraphs of the graph is required, which is provided by random graphs with sufficient edges.

On a final note the number of infected nodes the graph is started with, as long as it is small compared to $n$, does not substantially effect the number of rounds required for convergence.
Figure 4. Epidemic Protocols with Varying Degrees of Connectivity

Figure 5. Random Graph with 2% Connectivity
Phase Clock

The phase clock protocol is also discussed in [1], and has a close relationship with the one-way epidemic protocol. It is, in fact, designed to determine whether a one-way epidemic protocol has completed with high probability. To do this, the phase clock protocol must be able to count $\Theta(n \log n)$ interactions well, or, based on our analysis above, it must be able to count $1.5 n \log n$ interactions.

Like the epidemic protocol, the phase clock protocol is one way. To converge it requires the leader be in a strongly connected component. To ensure this occurs we again only consider undirected connected graphs.

The precise details of the protocol are simple. The state of each node is an integer value representing the ‘phase’ that node is in. When an initiator node contacts a non-leader responder, the responder updates his phase to the max of his phase and the initiators. If the responder is the leader he checks if the initiator has the same phase he does, and if so he moves to the next phase. In pseudocode:

\[
Q = \{\text{int phase, bool leader}\}
\]

Transition Function:

IF (LeaderResp)

\[
\begin{align*}
\text{IF (Phase}_{\text{Init}} == \text{Phase}_{\text{Resp}}) & \text{ Phase}_{\text{Resp}}++ \\
\text{ELSE} & \\
\text{IF(Phase}_{\text{Init}} > \text{Phase}_{\text{Resp}}) & \text{ Phase}_{\text{Resp}} = \text{Phase}_{\text{Init}}
\end{align*}
\]

For a version of this protocol which operates in bounded space see [1], the alterations are simple and only relate to the space bound, so they are not covered here. The phase clock protocol accepts as input whether a node is a leader or not, thus it starts with a leader. The leader’s phase is set to 1, while all other nodes are set to phase 0.

A new phase occurs each time the leader is contacted by a node with the same phase he has. With high probability a new phase doesn’t begin for $\Theta(n \log n)$ interactions[1], but to determine with high probability whether an epidemic has terminated multiple phases are required. A constant, $m$, is used so that a new ‘round’ only begins after $m$ phases have passed. The choice of $m$ alters the probability an epidemic has had time to complete itself and is independent of $n$. Unfortunately, while [1] has a thorough analysis which shows the existence of such a constant, it doesn’t reveal its value.

Figure 6 shows 100 trials of the phase clock protocol running for 100,000 interactions over 500 nodes. The blue dots are individual trials, while the pink dots are fitted line representing a linear function. In this case, the function is $y = x / 3390$. Since the phase clock protocol does not converge it is run for a set number of interactions and its progress recorded.

It is important that a linear function be a good approximation for the number of phases versus the number of interactions in the phase clock protocol. First, it fits our intuition. Each phase of the protocol should take roughly as many interactions as the one before it, accounting for the variance of a random scheduler. Consider the first phase, the leader propagates his value, which is propagated in turn. The 0 values held by other nodes at the start are never propagated. So, too, in phase $p$, values less than $p$ are never propagated to a node with phase $p$. When a new phase begins it can be seen to be identical to that first phase from the perspective of the leader and the set of nodes in phase $p$. Thus, we should expect any phase to take more or less the same number of interactions as the first phase.

Second, if the number of interactions per phase varied then it would be difficult, if not impossible, to pick an appropriate $m$ to determine when an epidemic finished. A constant number of interactions per phase, relative to $n$ and accounting for chance, is exactly what is called for.

Figure 7 shows that regardless of the number of nodes, each phase requires a constant number of interactions as a function of $n$. The tan values represent averages of the number of interactions required to reach each phase. They are shown for comparison purposes with the linear function. The same markers appear on Figure 8. Even from the naked eye their deviance from the average is tiny.

Extrapolating from the linear values required to create the pink lines on Figures 6 through 8 tells us that the gap between phases is $\approx 3/4 n \log n$. While the variance of the trials makes it difficult to calculate this constant exactly, using this equation to calculate best fit lines results in a n average deviance of only 3.2% from the actual lines of best fit. Most of this deviance is from the high variance trials, too.

From the previous subsection we know that $1.5 n \log n$ is a reasonable bound on the number of interactions required for the epidemic protocol to complete, as long as there are sufficient edges in the graph. To test our bound for the phase clock protocol we can set $m = 2$ and see with what percent the phase clock counts a round before the epidemic protocol can converge. If our constants are correct the result should be close to .5.

And, in fact, this is what’s found. On 500 nodes, for example, the average number of interactions for the phase clock protocol to reach $m = 2$ phases is 7004, while the average number of interactions for an epidemic to
complete in that circumstance is 6847, over 100 trials in both cases. Of the 100 trials of the phase clock, 48 of them reached phase 2 before 6847 interactions, while 52 did not. While it is difficult to make definitive statements given the high variance of the individual trials, putting these facts together strongly implies these constants are correct. Luckily, this more anecdotal data is supported by the best fit lines found in the figures.
Figure 7. Phase Clock Protocol on 300 Nodes

Figure 8. Phase Clock Protocol on 100 Nodes
Cancellation
The cancellation protocol is discussed in [1] and is described as follows:

\[ Q = \{ (0, 0), (1, 0), (0, 1) \} \]

Transition Function:

\[ (1, 0), (0, 1) \rightarrow (0, 0), (0, 0) \]
\[ (0, 1), (1, 0) \rightarrow (0, 0), (0, 0) \]

The cancellation protocol is defined as converging when only \((1, 0)\) and \((0, 0)\) or only \((0, 1)\) and \((0, 0)\) nodes remain. This requires a complete graph, and so our discussion will be constrained to complete graphs. The protocol accepts the state of each node as input.

Like the epidemic protocol, but unlike the phase clock protocol, the cancellation protocol is most interesting when the last few salient nodes remain. Consider an input set in which exactly half of the nodes are given the state \((1, 0)\), and the other half given \((0, 1)\). The convergence of this protocol will leave all nodes in the state \((0, 0)\), but it requires that each node find a ‘partner’ from the opposing set. While this is easy at first it becomes increasingly difficult as more nodes assume the state \((0, 0)\).

Before examining the figures for this protocol, it may be elucidating to state that our expectation for this protocol is that it will quickly ‘cancel’ many nodes until only a few nodes remain in either the state \((1, 0)\) or \((0, 0)\). It will then take an exceedingly long time to cancel those nodes. This differs from the epidemic protocol in that there is no start-up period in which few nodes are cancelled.

This difference is because in this protocol half the nodes are, essentially, starting off as ‘infected.’ Yet this protocol is not faster than the epidemic protocol. It is, in fact, \(\Theta(n^2)\) [1]. The reason for this is that the last few nodes to be cancelled in this protocol are much worse than those in the epidemic case. This can be seen experimentally in Figure 9.

While the graph is difficult to read as it comes close to convergence, between 60,000 rounds and 70,000 rounds the average trial (of 10 trials) cancelled zero new nodes. Between 0 rounds and 10,000 rounds, in contrast, the average trial run cancelled 477 nodes of 500. Between 0 and 100 the average trial cancelled 336.

This may seem unbelievable, but some simple math shows why it is the case that the final nodes are incredibly difficult to cancel. Take the last two out of 500 nodes in the complete graph with values \((1, 0)\) and \((0, 1)\) respectively. There is only one edge that can be selected to cancel them—the edge between them. In the complete graph on 500 nodes, however, there are \(500 \times 499 = 249,500\) edges. The chance of that one particular edge being selected on any round is \(.0004\%\)!

This protocol clearly demonstrates the importance of the cut between salient sets of nodes in these population protocols. It also suggests that pragmatic protocols in this model should require a high degree of accuracy, but not total convergence. When the last few nodes prevent an protocol from converging for a substantial proportion of its total runtime it would seem more useful to develop a method to exclude them.

Figure 10 shows the total run-time of the cancellation protocol based on the number of nodes. The blue points represent trial runs, while the tan triangles are placed at the average number of rounds required for convergence for a graph at each size. The pink line is merely \(y = x^2\), which fits fairly well. It’s deviation is too high to claim this is an appropriate constant, but it’s illustrative that the \(\Theta(n^2)\) seems correct.

Distance-2 Coloring
A more advanced and useful protocol, distance-2 coloring allows nodes to uniquely name their neighbors. It is described in detail in [3].

The state of each node is a boolean array of color values and a color. This protocol depends on knowing the maximum degree of the graph beforehand, as the color value array is large enough to have a spot for every color that is needed to distance-2 color the graph. This value is \(d(d-1)+1\), where \(d\) = maximum degree, since this is the most nodes at distance two a particular node might have. The array of color values starts initialized to false, and every node begins the protocol with color zero. The transition function is as follows:

\[ \begin{align*}
\text{IF } Q\text{init}[\text{Color resp}] \neq Q\text{resp}[\text{Color init}] & \quad \text{Color init} = (\text{Color init} + 1) \mod (d(d-1)+1) \\
& \quad Q\text{init}[\text{Color resp}] = Q\text{resp}[\text{Color init}] \\
\text{ELSE} & \quad Q\text{init}[\text{Color resp}] = \neg Q\text{init}[\text{Color resp}] \\
& \quad Q\text{resp}[\text{Color init}] = \neg Q\text{resp}[\text{Color init}]
\end{align*} \]

Unlike the other protocols discussed here, distance-2 coloring does not require a connected graph. However, cases in which the graph is not connected are not particularly interesting, as they are equivalent to coloring each component separately, then adding the number of rounds it took together.

Also unlike the other protocols, distance-2 coloring deals in constraints. Yet, despite this, distance-2 coloring’s run time is best determined by the density of edges across the
graph, just as is the epidemic protocol’s. But, the reason
this property governs distance-2 coloring’s runtime is
different. In the epidemic protocol a non-uniform density
do not edges could cause the epidemic to become stuck in a
high density subgraph with few routes out to the larger
uninfected subgraph. In the case of distance-2 coloring
an increase in density means an increase in the
constraints of color choice at each node.

Figure 11 shows a simple chart of the number of rounds
versus the number of edges required for the protocol to
run to convergence over 100 nodes. Figure 12 shows the
same date, plus the data from the protocol running to
convergence over 200 nodes. The data from the 200
node trials is in pink.

It is obvious that the number of edges in the graph is not
a good determiner of the run time of the protocol. Figure
13 charts the number of rounds required for convergence
versus the density (number of edges divided by number of
nodes) of the same trials. The data now fits coherently,
although it should be noted that the trials from the
graphs with 200 nodes still require more rounds to run to
collision. Even though they have the same density as
the smaller graph, they have more nodes to color. This
likely explains the difference, although it cannot be
conclusively stated either way at this point.

That the size of the problem makes such a tiny difference
implies that the constraints which govern protocols in
this model are most important. Often size and the
number of constraints go hand in hand, yet that is not
always the case. If a network of nodes using this model
were to be deployed they should carefully consider the
trade-offs inherent in adding edges when those edges
introduce new constraints to the graph.

CONCLUSIONS
Simulating theoretical research is a dangerous task. Theory,
being hard and pure, is not refutable. Trials, on the
other hand, can be misleading; especially when they
are subject to the high levels of variation possible in this
model of computation.

Yet an understanding of the way these protocols function
is informative to both theory and practice. The low
constants on the phase clock and epidemic protocols are
exciting since they open the door to pragmatic work. The
obscene number of rounds required to make a few nodes
in the cancellation protocol converge implies pragmatic
protocols in this model may have to sacrifice total
convergence. Finally, the understanding that the
complexity of the graph is what matters, more than its
size or connectivity, is key. The construction of future
protocols will likely rely upon this fact, and if networks
using this model are to be constructed this point will,
indeed must, inform their topology.

Considering that any real network will likely want to run
a series of protocols, the results regarding the one-way
epidemic protocol running well with few edges in the
graph implies that a network can be constructed which
will permit that protocol and the distance-2 coloring
protocol to run well. On the surface these protocols
would seem to require different topologies to perform
optimally, but luckily that is not the case.

Aside from the speculative fruits of this labor
determining solid constants for epidemics and the phase
clock allows those protocols to be employed. While
previously proven that there existed a constant which
would lower make the phase clock failing exponentially
unlikely, knowing that constant is comforting.

While it is unfortunate that the most complex protocol
covered in this paper, distance-2 coloring, turned out to
yield so little after so much work, the material pouring
forth from the simplest and most fundamental of
protocols employed by this model is fascinating. To think
that uniform density of edges across subgraphs would
matter more than connectivity is astounding. True, it is
perhaps intuitive when working closely in the mindset of
the population protocol model, for an outsider it proved a
discovery.

FUTURE WORK
While there’s near limitless work to be done, particularly
on leader election or possibly simulating Presburger
predicates, the most immediate work suggested from this
is calculating cuts during the running of the epidemic
protocol and constructing a measure of progress for the
distance-2 coloring protocol while it is in progress. These
things are not difficult to do in theory, but the current
implementation of the simulator cannot handle that kind
of computation. Without it we are left with correlation
and conjecture regarding the inner workings of the
epidemic protocol and the distance-2 coloring remains a
black box. I do not imagine its last few nodes are as bad
as the cancellation protocol, but its runtime does appear
to be quadratic.

ACKNOWLEDGMENTS
Thanks to Professor Aspnes for his patience and research
and to Professor Angluin for her advice and direction.

REFERENCES
1. Dana Angluin, James Aspnes and David Eisenstat.
Fast Computation by Population Protocols With a
Leader. In Distributed Computing, 20th International
Symposium, DISC 2006, 61-75.

2. Dana Angluin, James Aspnes, Melody Chan, Michael
J. Fischer, Hong Jiang and Rene Peralta. Stably
computable properties of network graphs. Available at
3. Dana Angluin, James Aspnes, Michael J. Fischer and Hong Jiang. Self-stabilizing population protocols. In
Figure 11. Distance-2 Coloring on 100 Nodes