The Chinese Generals Problem

Edwin Soedarmadji
California Institute of Technology
MC 136-93, Pasadena, CA 91125, USA
edwin@systems.caltech.edu

***Abstract*—To achieve higher reliability, safety, and fault-tolerance, many mission-critical detection and decision systems implement consensus algorithms that force the systems' underlying sensor networks to reach the states of consensus and unanimous decision among the sensor nodes. Most consensus algorithms presented in the literature utilize local averaging (for continuous values) and majority voting (for discrete values) operators combined with iterative message passing and other similar nearest-neighbor information propagation schemes.**

Although very simple to implement, such schemes can be very prone to noise because individual detection and decision errors can be amplified and propagated many times throughout the network. For this reason, in this paper we propose a novel consensus algorithm for binary systems that requires each sensor node to participate in message propagation only if its input exceeds a predetermined threshold. This algorithm is a solution to what we call *The Chinese Generals Problem*, a wide generalization of *The Byzantine Generals Problem* by Lamport *et al* [1]. The threshold function used in the algorithm leads to an adjustable network-wide threshold level that defines the minimum number of nodes initially reporting positive detection required in order for all the nodes to reach the correct consensus.

I. Introduction

A sensor network is built from a large number of sensor nodes distributed over an area where measurement of a specific phenomena or detection of a particular event is desired [2], [3]. Such a network is radically different from traditional networks in that its performance is largely determined by the density of its sensor nodes rather than their precise locations and interconnection topology. For this reason, sensor networks are often deployed in environmentally hostile areas where precise placement and configuration is prohibitively difficult. Operating in such areas, a sensor network is expected to experience a higher rate of node and link failures.

To successfully integrate any sensor network into a mission-critical detection and decision system, a designer needs to devise a mechanism that effectively addresses and mitigates these failures. For simplicity, in this paper we restrict our discussion to binary detection and decision systems.

In a binary detection system, all sensor nodes collectively attempt to ascertain the existence or absence of a particular event of common interest. In a binary decision system, measurement results from all sensor nodes are used to decide between two possible course of actions. Naturally, since the sensors may have different positions, operating environment, and operating parameters, they may end up with different conclusions on whether or not an event is detected, and if it is, on the most appropriate decision to make.

In one extreme scenario, the entire network delegates the decision making task to a single control node, which may or may not be a sensor node. Although appealing in its simplicity, this delegation effectively negates any desirable fault-tolerant properties that are inherent to sensor networks. Using this approach, the control node now becomes essentially a single point of failure of the entire network in several different ways.

First, either the node has to gather measurement results from all the sensor nodes, or all the sensor nodes have to reach the control node to transfer the measurement results. Second, even if we assume that all measurement results successfully reach the control node, they may not be all in agreement with one another. To reconcile the disagreement, the control node has to administer a (majority) voting procedure, which itself is also subject to failure, before finally committing to a decision.

Following the arguments in [4], we can say that if the control node is intended to be the one and only consumer of information produced by the sensor nodes, i.e., no other process or system relies on the control node's output, then from its point of view, the only relevant vulnerabilities are those related to routing and communication from the sensor nodes. However, if the control node affects other upstream processes, or if the network employ many control nodes for robustness, then processes at each control node become single-point network points of failure and sources for discrepancies.

The solution is to implement a *distributed voting* mechanism where every sensor node in the system aggregates information from its neighboring nodes, administers a local voting procedure, and waits for the next round of information gathering. First introduced by Lamport in 1982, *The Byzantine Generals Problem* [1] (BGP) addresses this need of redistributing the voting process. A *Byzantine Agreement* is reached when all operational sensor nodes agree on a common detection or an identical decision. At this stage, all operational sensor nodes carry the same information, and any upstream process can gather this information by sampling from any one of them.

Distributed voting cannot be used in systems that measures and chooses their values from the set of real numbers. Sensor networks in such systems use a *distributed averaging* mechanism to combine local information in the presence of noise, delay, and unstable topology [5]–[7], [8], [9].

As an alternative to voting and averaging, we propose a new aggregation mechanism based on thresholding. Although in this paper we restrict our analysis to binary threshold functions, the concepts and techniques introduced herein can be extended to threshold functions operating on real numbers.

This paper is organized as follows. In Section II the *Byzantine Generals Problem* (BGP) and one of its relevant variants are reviewed and compared to the proposed *Chinese General Problem* (CGP). In our attempt to clarify the notations and concepts of the CGP, we provide a detailed example in Section III. This example is then used as the foundation on which we present our main results in Section IV.

II. The Chinese Generals Problem

In their seminal 1982 paper, Lamport et al [1] introduced and analyzed the problem of reliable message synchronization in an otherwise unreliable distributed system, presented abstractly in terms of a group of Albanian Generals attempting to coordinate a synchronized action using messengers, and in the presence of treacherous generals amongst them. To obtain the impossibility theorem and other results in their paper, the problem is reduced into the one involving three Byzantine Generals, each of which representing an ensemble of Albanian generals and possibly some traitors. This reduced problem is called *The Byzantine Generals Problem*. Since its introduction, the problem has been studied by many researchers in the area of fault tolerant distributed computation [10]–[18].

From a large number of BGP variants that have been developed since, two variants are relevant to our paper. Wang et al [10] and Babaoglu et al [11], [12] generalized the BGP by not requiring a full network connectivity and by modifying the definition of a faulty component, respectively. In this paper, we further extend the BGP in the same directions as the previous two generalizations by relaxing even more assumptions.

Before discussing our generalization to the BGP, perhaps it is beneficial to first discuss what the BGP really is. Using the summary from [10], we can state that the BGP assumes the following: (1) there are S generals of which at most F generals may either maliciously or inadvertently change the received message before passing the message to the other generals; (2) all generals have direct access to each other through the use of messengers, i.e., the communication network is fully connected; (3) the sender can be identified from either the message or the messenger; (4) there is one commanding general who broadcasts an initial message v_0 in the beginning of the campaign and finally, (5) messages are modified only by the generals, but not by their messengers. The BGP is solved by developing optimal algorithms that can achieve the so-called *Byzantine Agreement* (BA): (BA1) All loyal generals agree on a common message v, and (BA2) if the commanding general is loyal, then v should be identical to the initial message v_0. In its original form, the BGP considers asynchronous communication between the generals.

The algorithm proven in [10] does not assume a fully connected network. This is a significant result as without the proposed relaxation, the network has to suffer a quadratic growth in the number of communication resources (I/O equipments, frequency, etc.) to guarantee that it can achieve a BA. Babaoglu extended the BGP by providing more specific failure modes that include crash and omission.

Crash is just another word for a permanent failure: once a general enters the crash mode, it can no longer send out its messengers to the other generals. In contrast, omission is just a temporary failure. A general ignores all the message it receives during the omission period, but continues to behave normally after the omission period is over.

Having described the BGP, we are ready to discuss our extensions. Using the next letter in the alphabet (after Albanian and Byzantine), we choose to call our extended problem *The Chinese Generals Problem* (CGP). First, some comments are due with regard to the similarities between the CGP and its superclass, the BGP. Both problems can be abstractly described as the problem faced by a team of generals who plan to commit a synchronized action by communicating via messengers. As such, solving the BGP and the CGP both involve developing an algorithm satisfying (BA1) and (BA2). Now we discuss the differences between the two.

In this paper, we introduce a preliminary version of the CGP that requires a synchronous communication (messenger) facility among the *memoriless* generals such that each message leaves the sender and arrives at the destination within one clock cycle. Consequently, in terms of synchronicity requirement, the CGP is more stringent than the BGP.

Apart from this real difference, an apparent difference comes from our conscious effort to restrict our analysis of the CGP to systems that exclusively use binary messages. While at first this might hint at a severe limitation of the CGP, Turpin and Coan [19] showed that BA can be reached in systems operating on multivalue messages that can be represented by k bits by running the appropriate binary BA algorithm k times.

The Chinese General Problem

Consider a group of Chinese Generals $G_i \in \mathcal{G}$, $i = 0 \dots S$ and the two possible actions A_0 and A_1 that they can take. Denote by $\mathcal{G}_1(t)$ the group of generals who choose A_1 at time $t \in \mathbb{N}$, and by $K(t)$ the number of such generals. The generals reach a consensus on A_1 iff there exists a certain time τ after which $K(t) \approx S$ (or $K(t) \approx 0$ if they instead agree on A_0). The CGP asks whether consensus is possible with the given algorithm described below, and if so, under what condition.

Starting from $t = 0$, each general G_i produces a binary bit $v_i(t)$ to indicate whether they choose A_0 or A_1. If $v_i(t) = 1$, then G_i dispatches M indistinguishable messengers to γ other generals $G_j \in \Gamma(G_i)$. The function Γ is such that we can construct an γ-regular graph G where $\mathcal{G}$ is the set of vertices of G, and $\{E_{ij}\}$ the set of edges of G connecting G_i to G_j whenever $G_j \in \Gamma(G_i)$. Let $L_i(t)$ be the number of messengers arriving at G_i at t. The update rule for $v_i(t)$ is:

$$v_i(t+1) = V(L_i(t)) = 1 \quad \text{iff} \quad L_i(t) \geq T. \tag{1}$$

Let us call T and V the threshold value and the threshold function, respectively. In this problem, the messenger is also allowed to not reach any of the γ generals and instead return to the originating general. Each messenger chooses the $\gamma + 1$ destinations with equal probability. Returning messengers are also counted as arrivals by the function $L_i(t)$. □

Compared to the majority voting function used in the BGP, V is more flexible because it allows a particular general G_i to still choose A_1 even if $L_i(t)$ indicates that it is not the majority opinion among the generals from whom information flows. In addition, threshold functions are also found in many different types of natural systems, the most famous being the ones found in neurons. In addition, the CGP does not assume that failures are contributed solely by the nodes (the generals) as assumed in [10]. In the CGP, communication links (the messengers) behave probabilistically, reflecting the relatively lower reliability that are often experienced in wireless and sensor networks. We will later show that the effect of node unreliability can be easily included in this model.

III. A Simple Example

Before proceeding to the next section for a full analysis of the CGP, let us recall that the definition includes the number of generals and their valency denoted by S and γ, respectively, the initial condition $K(0)$ which counts the number of generals deciding on A_1, and the two tunable parameters M and T denoting the number of messengers dispatched and the corresponding threshold value, respectively.

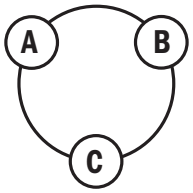

Fig. 1. Three generals

In this section, we analyze a simple example of a CGP where we use fixed values of $S = 3$, $\gamma = 1$, $M = 2$, and $T = 2$. The problem involves three generals labeled A, B, and C, linked in a simple triangular network shown in figure 1 above. Each general can send either zero or two messengers, and with equal probability each messenger can either go to the neighboring general in the clockwise direction, or stay.

For $\boldsymbol{v}(0) = \{v_A(0)\, v_B(0)\, v_C(0)\}$, we have eight possible configurations that can be written as binary strings: $\{000\}$, $\{001\}$, $\{010\}$, $\{100\}$, $\{011\}$, $\{101\}$, $\{110\}$, and $\{111\}$. All of them are equally likely. Notice that we sort the eight configurations according to the number of 1's in their strings. The reason is quite simple. The triangle in Figure 1 is symmetrical, which means configurations having the same number of 1's in their strings should be mathematically identical.

Let us consider the case where two generals, say A and B, initially decide on A_1 and dispatch two messengers each. Although the messengers are identical, for illustration purpose, we label the messengers at A's and B's positions with a_1, a_2, b_1, and b_2, and denote their positions at $t = 0$ by:

$$a_1a_2.b_1b_2.o$$

Messengers currently at A's, B's, and C's positions are written before the first dot, between the dots, and after the second dot, respectively, while an o indicates no messenger.

There are 16 possible ways the messengers can arrive at their destinations. Of course, with the exception of their labels, these configurations are practically identical to the ones produced in the cases where generals B and C, or C and A decide on A_1. Using the same notation, they are listed below.

$a_1a_2.b_1b_2.o$	$a_1a_2.b_1.b_2$	$a_1a_2.b_2.b_1$	$a_1a_2.o.b_1b_2$
$a_1.a_2b_1b_2.o$	$a_1.a_2b_1.b_2$	$a_1.a_2b_2.b_1$	$a_1.a_2.b_1b_2$
$a_2.a_1b_1b_2.o$	$a_2.a_1b_1.b_2$	$a_2.a_1b_2.b_1$	$a_2.a_1.b_1b_2$
$o.a_1a_2b_1b_2.o$	$o.a_1a_2b_1.b_2$	$o.a_1a_2b_2.b_1$	$o.a_1a_2.b_1b_2$

To determine $\boldsymbol{v}(t)$ at $t = 1$ if $\boldsymbol{v}(0) = \{110\}$), we have to compute $\boldsymbol{L}(0) = \{L_A(0), L_B(0), L_C(0)\}$ from the above before applying the threshold function V.

First, we will remove the distinction between all messengers arriving at the same general. To do this, we label the messengers currently at A's, B's, and C's with a, b, and c. As a shorthand, we denote aa by a^2 (and likewise with b). In this notation, the dots and o's are no longer needed:

a^2b^2	a^2bc	a^2bc	a^2c^2
ab^3	ab^2c	ab^2c	abc^2
ab^3	ab^2c	ab^2c	abc^2
b^4	b^3c	b^3c	b^2c^2

or, algebraically, as a polynomial where each term and its coefficient corresponds to a configuration and its multiplicities

$$\begin{aligned} F_{2a} &= b^4 + a^2b^2 + b^2c^2 + a^2c^2 \\ &= 2a^2bc + 4ab^2c + 2abc^2 + 2b^3c + 2ab^3 \end{aligned} \tag{2}$$

At $t = 0$, the configuration is always a^2b^2 because $\mathcal{G}_1(0) = \{A, B\}$. Suppose after the first round, the configuration becomes ab^2c. F_{2a} contains abundant information: there are four ways to achieve this configuration, and the value of $\boldsymbol{L}(0)$ is $\{\deg a\ \deg b\ \deg c\}$, which is $\{121\}$. With a little more work, we can get even more information: applying the threshold function V with $T = 2$ on $\boldsymbol{L}(0)$ gives us $\boldsymbol{v}(1) = \{010\}$ (an element of this string is a zero if the degree of the corresponding variable is less than T, and a one otherwise). In addition, we can also compute $K(0)$ and $K(1)$ by counting the numbers of ones in $\boldsymbol{v}(0)$ and $\boldsymbol{v}(1)$, which are 2 and 1, respectively. Calculating $K(t)$ is of interest to us because it shows the evolution of $\mathcal{G}_1(t)$.

Of course, in reality we can only compute an average value of $K(1)$ because the four messengers could just as well choose configurations other than ab^2c. The formula for the average is:

$$K(1) = \frac{\sum_i K_i(1)}{\kappa} \tag{3}$$

where the index i runs over all sixteen possible configurations at $t = 1$. In the numerator summand, the function $K_i(t)$ counts the number of ones in $\boldsymbol{v}_i(t)$ from a configuration i at time t. The denumerator κ is simply the total number of configurations, which in our present case is sixteen.

Having discussed $\mathcal{G}_1(0) = \{A, B\}$, we can now consider the other possible memberships of $\mathcal{G}_1(0)$: $\{\varnothing\}$, $\{A\}$, $\{B\}$, $\{C\}$, $\{A, B\}$, $\{A, C\}$, $\{B, C\}$, $\{A, B, C\}$. To do this, we need a polynomial that is more general than F_{2a}.

Consider the following polynomial $F(a,b,c;z)$ (or $F(z)$, for short). The coefficient F_k of z^k captures all the possible configurations given that $|\mathcal{G}_1(0)| = k$, i.e., k generals initially deciding on A_1. Obviously, $F(z)$ is more general than F_{2a} as all the configurations in F_{2a} can also be found in F_2.

$$\begin{aligned}
F(z) &= (1+(a+b)^2 z)\,(1+(b+c)^2 z)\,(1+(c+a)^2 z)\\
&= 1 + F_1 z + F_2 z^2 + F_3 z^3\\
F_k &= [z^k]F(z) = \tfrac{1}{k!}\tfrac{\partial^k}{\partial z^k}F(z)\big|_{z=0}\\
F_1 &= 2ab + 2bc + 2ca + 2a^2 + 2b^2 + 2c^2\\
F_2 &= a^4 + b^4 + c^4\\
&\quad + 2ab^3 + 2bc^3 + 2ca^3 + 2a^3b + 2b^3c + 2c^3a\\
&\quad + 3a^2b^2 + 3b^2c^2 + 3c^2a^2 + 8a^2bc + 8ab^2c + 8abc^2\\
F_3 &= 10a^2b^2c^2 + 2a^3b^3 + 2b^3c^3 + 2a^3c^3\\
&\quad + 6a^3b^2c + 6a^3bc^2 + 6a^2b^3c + 6a^2bc^3\\
&\quad + 6ab^3c^2 + 6ab^2c^3 + 2a^4bc + 2ab^4c + 2abc^4\\
&\quad + a^4b^2 + a^2b^4 + b^4c^2 + b^2c^4 + c^4a^2 + c^2a^4 \qquad \text{(X1)}
\end{aligned}$$

Let us first define F_{kl} as the number of l-th power of a, b, and c found in the monomials (i.e., terms, or configurations) of F_k. The index k restricts our count only to F_k, which is described above, while l restricts the count to only those generals having exactly l messengers: $\boldsymbol{L}_i(0) = l$.

For example, in F_2, the forms a^2, b^2, or c^2 are encountered 42 times (in the terms shown below) and hence $F_{22} = 42$:

$$3a^2b^2 + 3b^2c^2 + 3c^2a^2 + 8a^2bc + 8ab^2c + 8abc^2$$

Having defined F_{kl}, we can now compute $K_k(1)$, which is the average value of $|\mathcal{G}_1(1)|$ over all configurations in F_k.

$$\begin{aligned}
K_1(1) &= F_{12}/\kappa_1 = 6/(3\cdot 2^2)\\
&= 6/12 = 0.5000\\
K_2(1) &= (F_{22} + F_{23} + F_{24})/\kappa_2\\
&= (42 + 12 + 3)/(3\cdot 2^4)\\
&= 57/48 = 1.1875\\
K_3(1) &= (F_{32} + F_{33} + F_{34})/\kappa_3\\
&= (72 + 48 + 12)/(1\cdot 2^6)\\
&= 132/64 = 2.0625,
\end{aligned}$$

or more generally, we can use ($x' \in \mathcal{G}\setminus x$):

$$\begin{aligned}
K_k(1) &= \frac{1}{\kappa_k}\sum\nolimits_{l\ge T}^{Mk} F_{kl} \qquad (4)\\
F_{kl} &= \sum\nolimits_{x\in\mathcal{G}} \tfrac{1}{k!}\tfrac{\partial^l}{\partial x^l}F(a,b,c;z)\big|_{x=0,x'=1}\\
\kappa_k &= \tbinom{S}{k}(\gamma+1)^{Mk}
\end{aligned}$$

Consider the case where all generals are in $\mathcal{G}_1(0)$. Not knowing each other's decisions, they send their messengers to notify their neighbors. From $K_k(1)$, we predict that $|\mathcal{G}_1(1)| \approx 2$, while $|\mathcal{G}_1(2)| \approx 1$, and finally, at $t = 3$, the generals no longer decide on A_1 as $|\mathcal{G}_1(3)| < 1$. Therefore, with the given M, T, γ, and S, a proper consensus reflecting the generals' initial observations cannot be reached.

In Section IV, we present the general results for γ-regular network for all possible parameter values and show that with an appropriate choice of parameters, a proper consensus can be reached. In preparation for these general results, let us first generalize our triangular network into a ring network G with S generals (shown below with eight generals).

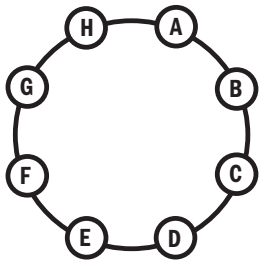

Fig. 2. The network G with eight generals

In our analysis of G, the values of M, S, and T are no longer fixed. As a result, we can no longer count the results manually and have to resort to the mathematical formalism of multivariate generating function (mgf). Let us denote the mgf for G by $F(\boldsymbol{x};z)$, with $\boldsymbol{x} = \{x_i\} = \{x_0, x_1, \ldots, x_{S-1}\}$ and $i \in \mathbb{N} \mod S$ (i.e., $i + S \equiv i \mod S$) indexing the S generals in $\mathcal{G}$. Denote by $[z^k]$ the coefficient of z^k and by $[x_i^l]$ the coefficient of x_i^l. As before, let i' denote any member of the set $\{0,\ldots,S-1\}\setminus i$.

$$\begin{aligned}
F(\boldsymbol{x};z) &= \textstyle\prod_{i=1}^{S}\left(1+(x_i+x_{i+1})^M z\right)\\
F_k(\boldsymbol{x}) &= [z^k]F(\boldsymbol{x};z) = \tfrac{1}{k!}\tfrac{\partial^k}{\partial z^k}F(z)\big|_{z=0}\\
F_{kl} &= \textstyle\sum_{i=1}^{S}[x_i^l]F_k(\boldsymbol{x}) \qquad (5)\\
&= S\tfrac{1}{k!}\tfrac{\partial^k}{\partial z^k}\tfrac{1}{l!}\tfrac{\partial^l}{\partial x_i^l}F(\boldsymbol{x};z)\big|_{x_i=0,x_{i'}=1}
\end{aligned}$$

The formulas for $K_k(t)$ and κ_k are the same as the ones found in (4). The summands in (5) are identical due to the symmetry of $F(\boldsymbol{x};z)$ with respect to $\boldsymbol{x}$. To derive the explicit formula for F_{kl}, suppose $l \ge 1$. Although counterintuitive, it is easier to start with $[x_i^l]$. Let $\alpha = (1+2^M z)^{S-2}$:

$$\begin{aligned}
F_l(z) &= S\,[x_i^l]F(\boldsymbol{x};z)\\
&= S\,\tfrac{1}{l!}\tfrac{\partial^l}{\partial x_i^l}F(\boldsymbol{x};z)\big|_{x_i=0,x_{i'}=1}\\
&= S\textstyle\prod_{i\notin\{1,S\}}\left(1+(x_i+x_{i+1})^M z\right)\big|_{x_i=1}\times\\
&\quad \tfrac{1}{l!}\tfrac{\partial^l}{\partial x_1^l}\left(1+(x_S+x_1)^M z\right)\left(1+(x_1+x_2)^M z\right)\big|_{x_1=0}\\
&= \alpha\,\tfrac{1}{l!}\tfrac{\partial^l}{\partial x_1^l}\left(1+(1+x_1)^M z\right)^2\big|_{x_1=0}\\
&= \alpha\,[x_1^l](1+2(1+x_1)^M z + (1+x_1)^{2M} z^2)\\
&= \alpha\left[\tbinom{2}{1}\tbinom{M}{l}(1)^{M-l}z + \tbinom{2}{2}\tbinom{2M}{l}(1)^{2M-l}z^2\right]\\
&= \textstyle\sum_{i=0}^{S-2}\tbinom{S-2}{i}2^{Mi}z^i\left[2\tbinom{M}{l}z + \tbinom{2M}{l}z^2\right]\\
F_{kl} &= [z^k]F_l(z) = \tfrac{1}{k!}\tfrac{\partial^k}{\partial z^k}F_l(z)\big|_{z=0} \qquad (6)\\
&= 2S\tbinom{M}{l}\tbinom{S-2}{k-1}2^{M(k-1)} + S\tbinom{2M}{l}\tbinom{S-2}{k-2}2^{M(k-2)}
\end{aligned}$$

At this point, recall that we haven't considered the case where $l = 0$, which requires a slightly different derivation.

The derivation for F_{k0}, is provided below. Again, we use the symmetry property of $F(\boldsymbol{x}; z)$. Note that at the first glance, the result for F_{k0} seems to be missing the $\binom{iM}{l}$ factor when compared to (6). However, recall that $\binom{x}{0} \equiv 1$ for all x.

$$\begin{aligned} F_{l=0}(z) &= \textstyle\sum_{i=1}^{S} F(\boldsymbol{x}; z)\big|_{x_i=0, x_{i'}=1} \\ &= S\, F(\boldsymbol{x}; z)\big|_{x_i=0, x_{i'}=1} \\ &= S \textstyle\prod_{i\in\{1,S\}} \left(1 + (x_i + x_{i+1})^M z\right) \big|_{x_i=0, x_{i'}=1} \\ &= S\,(1+2^M z)^{S-2}(1+z)^2 \\ F_{k0} &= S\,[z^k](1+2^M z)^{S-2}(1+z)^2 \qquad (7) \\ &= S\,\tfrac{1}{k!}\tfrac{\partial^k}{\partial z^k} \textstyle\sum_{i=0}^{S-2} \binom{S-2}{i} 2^{Mi} z^i (1+2z+z^2)\big|_{z=0} \\ &= S \textstyle\sum_{i=0}^{2} \binom{2}{i}\binom{S-2}{k-i} 2^{M(k-i)} \end{aligned}$$

This last observation suggests a formula for F_{kl} that is valid for $\gamma = 1$ and all values of k and l, that can be used in (4):

$$F_{kl} = \sum_{i=0}^{2} S \binom{iM}{l}\binom{2}{i}\binom{S-2}{k-i} 2^{M(k-i)}$$

To conclude our analysis on this example, we consider the other extreme value for λ. Now, instead of setting $\lambda = 1$ and working with a ring network, we set $\lambda = S-1$ and work with a complete graph, which is somewhat simpler to analyze:

$$\begin{aligned} F(\boldsymbol{x}; z) =& \textstyle\prod_{i=1}^{S} (1 + (\sum_{j=1}^{S} x_j)^M z) \\ =& (1 + (\textstyle\sum_{j=1}^{S} x_j)^M z)^S \\ F_l(z) =& S\,[x^l](1 + (S-1+x)^M z)^S \\ =& S\,[x^l] \textstyle\sum_{i=0}^{S} \binom{S}{i} (S-1+x)^{Mi} z^i \\ =& S\,[x^l] \textstyle\sum_{i=0}^{S} \binom{S}{i} \sum_{j=0}^{Mi} \binom{Mi}{j} (S-1)^{Mi-j} x^j z^i \\ =& S \textstyle\sum_{i=0}^{S} \binom{S}{i}\binom{Mi}{l} (S-1)^{Mi-l} z^i \\ F_{kl} =& S\,[z^k] \textstyle\sum_{i=0}^{S} \binom{S}{i}\binom{Mi}{l} (S-1)^{Mi-l} z^i \qquad (8) \\ =& S \binom{S}{k}\binom{Mk}{l} (S-1)^{Mk-l} \end{aligned}$$

As we mentioned previously, calculating $K_k(t)$ is important because the function allows us to learn about the evolution of $\mathcal{G}_1(t)$ and whether consensus is possible under the threshold function V. In our analysis of the triangular network, we manually calculated $K_k(t)$ for different values of k and learned that a proper consensus is not possible. Obviously, for the ring and complete network with variable parameters, a manual and exhaustive analysis of $K_k(t)$ is not an option.

However, all is not lost. Consider Mk, the number of messengers dispatched by k generals in $\mathcal{G}_1$, and $\lambda = \frac{Mk}{S}$. If we fix λ and $Mk \gg 1$, (8) becomes simplified and the qualitative behaviors of $K_k(t)$ and the consensus become clear.

$$P_{kl} = \frac{F_{kl}}{S\,\kappa_k} = \frac{S\binom{S}{k}\binom{Mk}{l}(S-1)^{Mk-l}}{S\binom{S}{k}S^{Mk}} \to \frac{\lambda^l}{l!}e^{\lambda}$$

$$K_k(1) = S \textstyle\sum_{l\geq T}^{Mk} P_{kl} \to S \sum_{l\geq T}^{Mk} \frac{\lambda^l}{l!} e^{\lambda} = S(1 - \frac{\Gamma(T,\lambda)}{(T-1)!})$$

Perhaps not surprisingly, for $Mk \gg 1$ and $\gamma = S-1$, each term in the summation in (4) is a Poisson density P_{kl} with parameter $\lambda = \frac{Mk}{S}$ and argument l, and thus $K_k(1)$ can be expressed in terms of the incomplete gamma function Γ.

Figure 3 is a plot of $K_k(t+1)$ against $K_k(t)$, which is actually the plot of $K_k(1)$ against $k = |\mathcal{G}_1(0)|$ for $M = 2$ (the lower curve) and $M = 4$ (the upper curve) compared to the diagonal line $K_k(t+1) = K_k(t)$. For both curves, $\gamma = S-1$, $T = 2$, and $S = 100$. These curves can be used to describe the time evolution of $K_k(t)$.

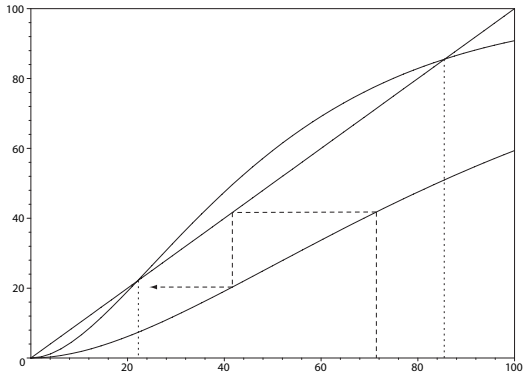

Fig. 3. The plot of $K_k(t+1)$ versus $K_k(t)$

For example, suppose $K_k(0) = 72$. For the $M = 2$ curve, $K_k(1) \approx 42$, and $K_k(2) \approx 20$, before eventually reaching $K_k(t) = 0$ (the dashed path with an arrowhead). In other words, all generals would eventually settle on A_0 no matter how many of them initially agreeing on A_1. In contrast, the $M = 4$ curve intersects the diagonal line at $k \approx 22$ and $k \approx 85$. These points define two possible steady-state behaviors of $K_k(t)$. If $K_k(0) < 22$, then as $t \to \infty$ $K_k(t)$ settles at the stable fixed point at $k = 0$. Otherwise, it settles at $k \approx 85$.

Denote by $F(k)$ the function that maps $K_k(t)$ to $K_k(t+1)$. The properties of Γ require that $F(k) \to 0$ as $k \to 0$ and $F(k) \to S$ as $t \to \infty$, and in addition, the slopes $\frac{\partial F(k)}{\partial k} \to \frac{1}{S}$ and $\frac{\partial F(k)}{\partial k} \to \frac{1}{S}$ as $t \to 0$ and $t \to \infty$, respectively. Due to space limitation, we do not analyze of the effects of changing M and T on $F(k)$, and proofs to the above claims. We do, however, provide in the next section a sketch of the analysis of errors caused by the generals themselves.

IV. General Result

Finally, we present the most general version of the CGP with no limits on any of the parameters. Shown in Figure 4(a) is one such network with $S = 9$ generals arranged in a 3×3 grid with $\gamma = 4$ such that each general G_i can send his messengers to the neighbors N_i to the east, west, north, and south (in addition to himself). The network wraps around on itself, so some neighbors are on the opposite side of the grid.

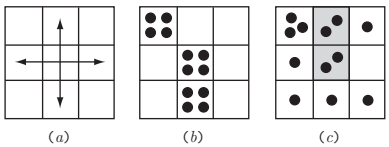

Fig. 4. A general γ-regular network

Figure 4(b) shows the $k = 3$ generals from $\mathcal{G}_1(0)$, each equipped with $M = 4$ messengers. Figure 4(c) shows one possible messenger configuration after redistribution. The two shaded cells mark the generals G_i such that $l = L_i(1) = 2$. As before, F_{kl} then computes the number of such cells in all possible configurations. We generalize our previous results and proof that the following is true ($L = \gamma + 1$):

$$F_{kl} = \sum_{i=0}^{L} S\tbinom{Mi}{l}\tbinom{L}{i}\tbinom{S-L}{k-i}(L-1)^{Mi-l}L^{M(k-i)} \qquad (9)$$

To proof the above equation, we use the same technique we used in the previous section. First, start with the mgf.

$$\begin{aligned}
F(\boldsymbol{x}; z) =& \textstyle\prod_{i=1}^{S}(1 + (\sum_{j\in N_i} x_j)^M z) \\
F_l(z) =& S\left[x_n^l\right] \textstyle\prod_{i=1}^{S}(1 + (\sum_{j\in N_i} x_j)^M z) \\
=& S\left[x_n^l\right] (1 + L^M z)^{S-L}(1 + (L - 1 + x_n)^M z)^L \\
=& S\,\tfrac{1}{l!}\tfrac{\partial^l}{\partial x_n^l}(1 + L^M z)^{S-L} f(x_n; z)\big|_{x_n=0} \\
f(x_n; z) =& (1 + (L - 1 + x_n)^M z)^L \\
=& \textstyle\sum_{i=0}^{L}\binom{L}{i} z^i \sum_{h=0}^{Mi}\binom{Mi}{h}(L-1)^{Mi-h}x_n^h
\end{aligned}$$

Now we substitute the above expressions into F_{kl}:

$$\begin{aligned}
F_{kl} &= S\left[z^k\right] (1 + L^M z)^{S-L}\tfrac{1}{l!}\tfrac{\partial^l}{\partial x_n^l} f(x_n; z)\big|_{x_n=0} \\
&= S\left[z^k\right] (1 + L^M z)^{S-L} \textstyle\sum_{i=0}^{L}\binom{L}{i} z^i \binom{Mi}{l}(L-1)^{Mi-l} \\
&= S \textstyle\sum_{i=0}^{L}\binom{L}{i}\binom{Mi}{l}(L-1)^{Mi-l}[z^k](1 + L^M z)^{S-L} z^i
\end{aligned}$$

Solving the last line gives us an expression identical to (9):

$$F_{kl} = \sum_{i=0}^{L} S\tbinom{L}{i}\tbinom{Mi}{l}\tbinom{S-L}{k-i}(L-1)^{Mi-l}L^{M(k-i)} \qquad (10)$$

The expression for F_{kl} from (10) can then be substituted into (4) to obtain a plot similar to Figure 3. We also note that all γ-regular graphs have the same mgf, and hence the same F_{kl}, whether or not they are isomorphic to each other. Figure 5 shows two non-isomorphic 4-regular octagons with identical F_{kl}. Aside from node labels, their mgf's are exactly the same.

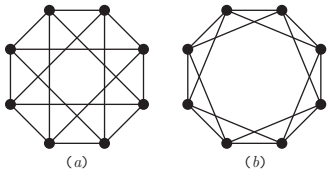

Fig. 5. Two non-isomorphic 4-regular octagonal network [20], [21]

Finally, we propose one way of modeling the case where the generals themselves commit decision errors. Suppose we have $k = |\mathcal{G}_1(t)|$. With probability p, each of the k generals may decide not to dispatch their messengers, and likewise, with probability p, each of the remaining $S - k$ generals may spontaneously decide to dispatch their messengers. Mathematically, this is equivalent to transforming the function $F(k)$ into $F'(k) = F(k) + p(S - k) - pk$.

An easy way to visualize this transformation is to imagine rotating $F(k)$ around a pivot located at $(\frac{S}{2}, F(k))$. Obviously, as a result, $F'(k)$ may or may not have any stable fixed point. However, the good news is that with a proper choice of parameters, $F'(k)$ can have two different fixed points with high and low values of k to represent two possible consensus on A_0 and A_1, respectively. As a general rule, it is desirable to separate these two fixed points as far as possible to improve the network resistance against spurious noise.

ACKNOWLEDGMENTS

This work was supported by the Caltech Lee Center for Advanced Networking and NSF Grant No. CCF-0514881. The author would also like to thank Prof. Robert J. McEliece for useful discussions and continued support for this research.

REFERENCES

[1] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem." *ACM Trans. Program. Lang. Syst. 4*, 3 (July), 382-401.
[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, August 2002.
[3] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, 2004.
[4] M. Raynal, *Distributed Algorithms and Protocols*, John Wiley, 1988.
[5] R. Olfati-Saber and R.M. Murray, "Consensus problems in networks of agents with switching topology and time-delays." *The IEEE Transactions on Automatic Control*, 49(9):1520-1533, September 2004.
[6] R. Olfati-Saber and R.M. Murray, "Consensus protocols for networks of dynamic agents". *Proc. of the Amer. Cont. Conf.*, 2:951-956, June 2003.
[7] R. Olfati-Saber and R.M. Murray, "Flocking with Obstacle Avoidance: Cooperation with Limited Communication in Mobile Networks." *Proc. of the 42nd IEEE Conf. on Dec. and Control*, 2:2022-2028, Dec. 2003.
[8] W. Ren and R.W. Beard, "Consensus of Information Under Dynamically Changing Interaction Topologies." *Proc. of the Amer. Cont. Conf.*, 2004.
[9] A. Tahbaz-Salehi and A. Jadbabaie, "On Consensus Over Random Networks", *Proc. of the 44th Annual Allerton Conference*, UIUC, Illinois, USA, Sept 2006.
[10] S.C. Wang, Y.H. Chin, and K.Q. Yan, "Byzantine Agreement in a Generalized Connected Network," *IEEE Trans. Par. and Dist. Systems.*, vol. 6, pp. 420–427, April 1995.
[11] Babaoglu 0. and Drummond R., "Streets of Byzantium: Network architectures for fast reliable broadcasts," *IEEE Trans. Software Eng.*, vol. SE-11, pp. 546–554, June 1985.
[12] 0. Babaoglu, P. Stephenson, and R. Drummond, "Reliable broadcasts and communication models: Tradeoffs and lower bounds," *Dist. Computing*, vol. 2, pp. 177–189, 1988.
[13] A. Bar-Noy, D. Dolev, C. Dwork, and R. Strong, "Shifting gears: Changing algorithms on the fly to expedite Byzantine Agreement," *Proc. Symp. Principles Dist. Computing*, 1987, pp. 42–51.
[14] K.M. Chandy and J. Misra, *Parallel Program Design: A Foundation*. Reading, MA: Addison-Wesley, 1988.
[15] D. Dolev and R. Reischuk, "Bounds on information exchange for Byzantine Agreement," *JACM*, vol. 32, no. 1, pp. 191–204, Jan. 1985.
[16] M. Fischer, "The consensus problem in unreliable distributed systems (a brief survey)," *Lecture notes in computer science, in Proc. 1983 Int. FCT-Con.* Borgholm, Sweden, Aug. 1983, pp. 127–140.
[17] B.M. McMillin and L.M. Ni, "Byzantine fault-tolerance through application oriented specification," *Proc. COMPASAC87*, 1987, pp. 347–353.
[18] R. Reischuk, "A new solution for the Byzantine generals problem," *IBM Res. Rep.*, RJ–3673, Computer Science, 1982.
[19] R. Turpin, and B. Coan, "Extending binary Byzantine agreement to multivalued Byzantine agreement", *Process, Lett.* 18, 1984, pp. 73-76.
[20] R.C. Read and R.J. Wilson, *An Atlas of Graphs*. Oxford, England: Oxford University Press, 1998.
[21] N.J.A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, Sequences A006820/M1617, A033301, and A033483.