“Not All At Once!” – A Generic Scheme for Estimating the Number of Affected Nodes While Avoiding Feedback Implosion

Reuven Cohen and Alexander Landau
Dept. of Computer Science
Technion
Israel

Abstract—We present a generic scheme for estimating the size of a group of nodes affected by the same event in a large-scale network, such as a grid, a sensor network or a wireless broadband access network, while receiving only a small number of feedback messages from this group. Using the proposed scheme, a centralized gateway analyzes the transmission times of these feedback messages, defines a likelihood function for them, and then uses the Newton-Raphson method to find the number of affected nodes for which this function is maximized. We present complete mathematical analysis for the precision of this algorithm and provide tight upper and lower bounds for the estimation error. These bounds allow us to improve the precision of our estimation, and to bring the estimation error very close to 0.

Index Terms—feedback implosion, management of big clusters, group size estimation

I. INTRODUCTION

In this paper we address a problem that arises in many modern networks, such as Grid networks, satellite networks, sensor networks and broadband access wireless networks. Such networks consist of thousands of end devices (nodes) that are controlled or managed by a single centralized gateway. From time to time the end nodes must send feedback messages to the gateway concerning local events about which the gateway has to be aware. While some events are only detected by a single node or a few nodes, some important events are likely to affect many nodes. The scheme proposed in this paper is useful when it is crucial that the gateway not only be informed about such events but also have a rough estimation of the number of affected nodes, without having each node send a message to the gateway.

The proposed scheme is called NATO! (Not All aT Once!). The main idea is that after the gateway announces a possible event, every affected node waits a random amount of time before sending a feedback RPRT (report) message. When the gateway receives enough RPRTs to estimate the number of affected nodes with good precision, it broadcasts a STOP message, telling the nodes that have not reported yet not to send their RPRTs.

The most important part of this paper is the development of a statistical analysis algorithm, to be employed by the gateway, for estimating the number of affected nodes. The estimation is based on the times at which the RPRTs are sent. This algorithm defines the likelihood function for the received RPRTs, and then uses the Newton-Raphson method to find the number of nodes for which this function is maximized. Using mathematical analysis, we provide tight upper and lower bounds on the estimation error. We show that this error is approximately $1/(N-1)$, where $N$ is the number of sent RPRTs, and it is always positive. We use this property to bring the estimation error very close to 0.

The rest of this paper is organized as follows. In Section II we present more detailed application scenarios along with related work. In Section III we present the estimation algorithm, which is the core of the proposed NATO! scheme. In Section IV we analyze the precision of this algorithm and find tight upper and lower bounds for the error. This analysis allows the error introduced by our estimation to be reduced and brought very close to 0 even if $N$ is very small compared to $r$. In Section V we analyze the effect of feedback losses on the precision of the estimation. In Section VI we conclude the paper.

II. APPLICATION SCENARIOS AND RELATED WORK

A. Application scenarios

The proposed NATO! scheme is suitable for networks and systems that fulfill the following requirements:

(R1) The network consists of thousands of end nodes reporting to a single centralized gateway. Having each affected node send a separate RPRT (report) message to the gateway would result in one of the following implosion effects: (a) insufficient network resources for forwarding the messages to the gateway; (b) insufficient gateway CPU resources for processing all these messages; (c) delayed gateway response to the event.

(R2) It is not enough that the sender knows about the event. In order to correctly respond to it, the sender also needs a good estimate of the number $r$ of nodes that have experienced this event.

(R3) There is a strict limit between the time the event takes place and the time the gateway needs to estimate the number of affected nodes.

(R4) The gateway is able to broadcast a START message and a STOP message to all of the nodes that might be affected by the event. The START message indicates
that RPRT messages should be sent and the STOP message indicates that no more RPRT messages should be sent. (R5) To ensure synchronous execution of the protocol, one of the following requirements must hold: (a) the gateway and the nodes share a global time, e.g., using a GPS; or (b) the delays from the gateway to all the nodes are almost equal, as in most of the single hop wireless/cellular networks when the base-station plays the role of the gateway; or (c) the gateway knows the delay $D_i$ to/from every node $i$.

We now present some application scenarios for which requirements R1-R5 hold, and present related work for each of them.

An important application scenario is the detection of denial-of-service (DoS) attacks [25] in sensor networks. Our NATO! scheme can discover many such possible attacks, especially when all sensors have directly wireless connectivity with the gateway. For instance, [18] describes an attack where the attacker prevents the sensors from receiving the gateway’s broadcast messages. Using the proposed NATO! scheme, the gateway can periodically estimate the number of sensors that are able to receive its broadcast messages without requiring each of them to send an individual response. Attacks on the physical and MAC layers of sensor networks, such as jamming attacks [25], can be efficiently discovered by NATO! in a similar way.

The NATO! scheme can also be used to discover a DoS attack in a 802.16-like mobile wireless network [13]. Possible attacks that prevent connectivity from many hosts to the base-station are described in [5]. In this context, the 802.16 base-station plays the role of NATO!‘s gateway. The base-station periodically invokes the scheme in order to ensure that the estimated number of responding nodes is roughly equal to the number of supposedly active nodes.

The third application we discuss is management of grid networks. As organizations deploy large, Internet-scale, computational and data grids, the necessity for systems that monitor and control vast amounts of available resources has become apparent. Such systems require a substantial amount of monitoring data to be collected for a variety of tasks such as fault detection, performance analysis, performance tuning, performance prediction, and scheduling [24], [26].

For scalability issues, such monitoring and control systems have a hierarchical structure. Still, a management entity is responsible for thousands of nodes, and it is essential to minimize the amount of control information sent by each. Because a grid consists of clusters of nodes, hundreds of nodes in the same cluster may be affected by an event. For instance, hundreds of farm PCs that are connected to the same storage will be affected when this storage fails. The NATO! scheme can substantially reduce the number of messages received by a grid management station or a grid local management station, in such cases. Even if the management station receives only 4-7 messages, it can still estimate the number of grid nodes affected by the event.

The fourth application we describe for the NATO! scheme is reliable multicast in broadcast wireless/cellular/satellite networks. A prominent feature of these technologies is the base-station’s ability to transmit a single copy of a packet to a huge group of receivers. In a typical FEC-based reliable multicast, the sender creates from each data block $K+n$ packets. To decode the data block, a receiver must receive any $K$ of these packets. In a hybrid FEC/ARQ-based scheme [1], [2], [11], [15], [20], [22], receivers that have not received enough packets correctly notify the sender, by means of a NACK message, and the sender transmits additional repair packets. The number of such repair rounds is usually limited by real-time considerations.

One way to use NATO! for FEC-based reliable multicast without ARQ is as follows. Once every time-out period (e.g., once a second), the sender invokes NATO! in order to estimate the loss distribution for the considered multicast group. That is, the sender estimates the number of nodes in this group that have lost $p\%$ of multicast packets since the previous time-out, for several relevant values of $p$. Using this information, the sender can determine the number $n$ of proactive repair packets that have to be transmitted in addition to the $K$ packets required for the decoding of every data block. This value of $n$ is used for the considered multicast group until the next time NATO! is invoked.

The last application scenario is the prevention of feedback implosion in sensor networks. Due to the requirement of NATO! that the gateway will initiate the transmission of RPRT messages by the nodes, we concentrate here on sensor networks where the gateway periodically asks the nodes to report about specific events, such as a local temperature that exceeds some threshold. Most papers that address this problem adopt the concept of data aggregation, e.g., [7], [12], [14], [23]. The idea is that similar data messages sent by multiple sources are aggregated by the network nodes. The aggregation depends on many factors, such as message content, identification, urgency, or the processing and storage (caching) capability of the intermediate nodes. Although useful, the data aggregation concept significantly increases the cost and complexity of the sensor nodes. Moreover, it renders the sensor network vulnerable to eavesdropping and data tampering [8], or requires security association between neighboring sensors. Therefore, solutions that avoid aggregation might be useful in many applications. The proposed NATO! scheme can solve the problem of feedback implosion without using data aggregation.

We are not aware of any paper that addresses requirements R1-R5 like the NATO! scheme proposed in this paper. Some works have addressed a slightly different problem: estimating the total number of receivers in a multicast group [3], [4], [17], [19]. These works take advantage of the strong correlation between successive measurements when the size of a multicast group is estimated. This correlation, used to reduce the cost of the estimation process, does not exist in the NATO! applications considered above.

B. Related work for the NATO! mathematical scheme

The authors of [4] discuss the $M/M/\infty$ model for receivers entering and exiting the multicast group. To avoid feedback implosion, not all the receivers send a message to the sender.
Rather, each one sends a message with a predefined probability \( p \). The sender uses \( p \) and other parameters to estimate the number of receivers. The authors of [3] extend [4] by relaxing several assumptions and using different filters on the received feedback messages. As noted earlier, our NATO! scheme differs from the one proposed in [3], [4] in that we do not assume any correlation between two successive measurements. Furthermore, in our approach, the gateway sets a hard limit on the number of feedback messages it is willing to receive.

The authors of [19] also employ a timer-based approach. However, in their scheme receivers should stop sending feedback messages after the first one is transmitted. In [10], the authors propose a probabilistic polling model for estimating the size of a multicast group. They consider a known probability distribution function and sends a feedback message when it expires. When the feedback message arrives at the sender, it broadcasts the next RFB, which also stops the receivers from sending additional responses to the previous RFB. The number of feedback messages received by the sender is therefore proportional to the length of the RTT.

As in NATO!, the authors of [16] also employ the concept of maximum likelihood. However, they do so for the sake of estimating the size of a multicast group. They consider a sender broadcasting an RFB (Request for Feedback) message to a group of receivers. Each receiver sets a random timer using a known probability distribution function and sends a feedback message when it expires. When the first feedback message arrives at the sender, it broadcasts the next RFB, which also stops the receivers from sending additional responses to the previous RFB. The number of feedback messages received by the sender is therefore proportional to the length of the RTT.

The fact that our paper and [16] address different problems is translated into the following differences:

- We overcome feedback implosion by limiting the number of response messages sent by receivers, while [16] limits the time during which these messages are sent.
- Running the algorithm of [16] in a network with a small RTT will result in a single response message being received by the sender. In contrast, in NATO! the number of response messages is mainly affected by the required estimation precision.
- In our scheme, each measurement of the size of the group of affected nodes is independent of previous measurements, while in [16] the results of previous measurements are taken into account.

NATO! is an important building block in each of the application scenarios considered above. Still, problems specific to each of these applications must be closely examined and addressed. For example, in the context of reliable broadcast in wireless/cellular/satellite networks, one needs to show how to implement NATO! when the response messages are subject to collisions on the shared uplink. This and other issues are addressed in [9].

III. THE ESTIMATION ALGORITHM OF NATO!

Let \( r \) be the number of affected nodes, namely, nodes that are supposed to send a response/report (RPRT) message to the gateway after receiving a START message. Our goal is to estimate this number while limiting the number of RPRTs sent by affected nodes. When a node discovers that it is affected by an event that has to be reported to the gateway, it invokes the following algorithm:

*Algorithm 1*: This algorithm is invoked by every affected node upon receiving a START(\( t_0 \)) message from the gateway, where \( t_0 \) is the time after which RPRT messages can be sent to the gateway.

- Choose a random timer in the range \([t_0, t_0 + T]\) using a known probability distribution function \( F \).
- If the timer expires before a STOP message is received from the gateway, send a RPRT to the gateway. Otherwise, do not send.

Following this algorithm, the gateway receives during \([t_0, t_0 + T]\) a number of RPRT messages. Let this number be \( N \). Let \( f \) be the probability density function of \( F \). Let \( X_1, \ldots, X_N \) be random variables denoting the transmission times of the \( N \) RPRTs. Without loss of generality, these random variables are assumed to be ordered in non-decreasing order such that \( X_1 \leq X_2 \leq \ldots \leq X_N \). Finally, let \( x_1, \ldots, x_N \) denote the exact values of \( X_1, \ldots, X_N \) in a specific experiment.

Without loss of generality, in the following analysis we assume that \( t_0 = 0 \). We use the maximum likelihood method to estimate \( r \). Let \( f_{X_1, X_2, \ldots, X_N}(x_1, x_2, \ldots, x_N) \) be the joint density function of \( X_1, X_2, \ldots, X_N \) given that the number of affected nodes is \( r \). This function is the probability density of the first \( N \) order statistics of distribution \( F \), for which it is known that [21]:

\[
f_{X_1, X_2, \ldots, X_r}(x_1, x_2, \ldots, x_r) dx_1 \ldots dx_r =
\]

\[
= P(X_1 \in (x_1, x_1 + dx_1), \ldots, X_r \in (x_r, x_r + dx_r)) =
\]

\[
= r! P(Y_1 \in (x_1, x_1 + dx_1), \ldots, Y_r \in (x_r, x_r + dx_r)) =
\]

\[
= r! (F(x_1 + dx_1) - F(x_1)) \ldots (F(x_r + dx_r) - F(x_r)) =
\]

\[
= r! f(x_1) \ldots f(x_r) dx_1 \ldots dx_r,
\]

where \( Y_1, \ldots, Y_r \) are independent random variables from distribution \( F \). Therefore,

\[
f_{X_1, X_2, \ldots, X_r}(x_1, x_2, \ldots, x_r) = r! f(x_1) \ldots f(x_r).
\]

In order to find the joint density of the first \( N \) \( X_i \)'s, we integrate over \( x_{N+1}, \ldots, x_r \):
Define the likelihood function \( L(r) \) to be
\[
L(r) = f_{X_1, X_2, \ldots, X_N}(x_1, x_2, \ldots, x_N).
\]

We now seek for the value of \( r \) that maximizes \( L(r) \). Such an \( r \) yields the maximum likelihood for getting the considered experiment’s outcome, \( x_1, \ldots, x_N \), and is therefore the most probable number of affected nodes. We find the maximum of \( L(r) \) by differentiation. Since \( L(r) \) is a product of other functions, it is hard to differentiate it directly. Since \( ln \) is a monotonically increasing function, \( L(r) \) gets its maximum for the same value of \( r \) as \( l(r) \), where
\[
l(r) = \ln L(r) = \ln \frac{r!}{(r-N)!} + \ln f(x_1) + \ldots + \ln f(x_N) + (r-N) \ln(1-F(x_N)) + (r-N+1) \ln(1-F(x_N)) + \text{const}.
\]

In this equation, \( \text{const} \) is a constant with respect to \( r \).

We now differentiate \( l(r) \) with respect to \( r \) and get
\[
l'(r) = \frac{1}{r} + \frac{1}{r-1} + \ldots + \frac{1}{r-N+1} + \ln(1-F(x_N)).
\]

Thus, in order to find the value of \( r \) which maximizes the likelihood function \( L(r) \), we need to find real values of \( r \) that satisfy the following equation:
\[
\frac{1}{r} + \frac{1}{r-1} + \ldots + \frac{1}{r-N+1} + \ln(1-F(x_N)) = 0. \quad (3)
\]

Proposition 1: From the \( N \) possible real solutions of Eq. 3, the one that maximizes \( L(r) \) is the maximum one.

Proof: \( L(r) \) and \( l(r) \) get their maximum at the same \( r \). Thus it is enough to show that \( l(r) \) gets its maximum at the maximum solution of Eq. 3.

Since for every \( r \)
\[
l''(r) = -\frac{1}{r^2} - \frac{1}{(r-1)^2} - \ldots - \frac{1}{(r-N+1)^2} < 0, \quad (4)
\]
then any real root of Eq. 3 is a local maximum of \( l(r) \). The global maximum is one of the local maxima, so it remains to find which of the local maxima gives the highest value of \( l \).

Substituting
\[
\ln(1-F(x_N)) = -\left(\frac{1}{r} + \frac{1}{r-1} + \ldots + \frac{1}{r-N+1}\right)
\]
from Eq. 3 into Eq. 2 yields:
\[
l(r^*) = \ln r^* + \ldots + \ln(r^* - N + 1) + \text{const}
\]
\[
= \ln r^* + \ldots + (r^* - N + 1) - 1 - \left(1 + \frac{1}{r^* - 1}\right) - \ldots - \left(1 + \frac{N - 1}{r^* - N + 1}\right) + \text{const}.
\]

This is a monotonically increasing function. Therefore, of all the roots \( r^* \) of Eq. 3, the one whose value is maximum will maximize both \( l(r) \) and \( L(r) \).

A practical method for solving Eq. 3 is as follows. Since the \( ln \) term is constant, the equation has the form \( 1 + \frac{1}{r+1} + \ldots + \frac{1}{r+N} + c = 0 \). This function has vertical asymptotes at points \( r = 0, 1, \ldots, N - 1 \). From Eq. 4 it follows that the function decreases monotonically at every interval \((i - 1, i), i = 1, \ldots, N - 1 \) and thus it has \( N - 1 \) roots at the interval \((0, N - 1) \). The function also decreases monotonically at the interval \((N - 1, \infty) \), and thus has its greatest root in this interval. This is the root we are seeking. To find it, the sender can employ the Newton-Raphson method. Given an equation \( h(x) = 0 \) where \( h \) is a continuously differentiable function and given a starting point \( x_0 \), near which the equation root is located, the method iteratively finds an approximation for the root with any desirable precision. On the \((n + 1)\)-th iteration, \( x_{n+1} = x_n - \frac{h(x_n)}{h'(x_n)} \), where \( h'(x) \) is the derivative of \( h(x) \). The idea is to find the tangent of \( h \) at \( x_n \) and to set \( x_{n+1} \) to the point where the tangent crosses the \( x \)-axis, thereby getting closer to the root. In our case, \( h \) is given in Eq. 3 and \( x_0 = N - 1 + \varepsilon \), where \( \varepsilon \) is small positive number. This starting point was chosen for two reasons. First, there must not be asymptotes between the starting point and the root (therefore \( x_0 > N - 1 \)). Second, there must not be asymptotes between any \( x_n \) and the root, and since the function is monotonically decreasing at the interval \((N - 1, \infty) \), it is implied that \( x_0 < \text{root} \). The whole process stops when \( |h(x_n)| \) gets sufficiently close to 0. In the simulation results, shown later, we stopped when \( |h(x_n)| < 0.001 \), which usually holds after 9-10 iterations. The value \( x_n \) for the last \((n)\)-th iteration is taken to be the solution of Eq. 3, namely, the estimated value of \( r \).

To conclude, the algorithm executed by the gateway for estimating the number of affected nodes \( r \) is as follows.

Algorithm 2: The gateway algorithm:

- Broadcast a START\((l_0)\) message to all possible affected nodes.
- When \( N \) RPRTs are received, broadcast/multicast a STOP message to all possible affected nodes.
- Use the Newton-Raphson method, as described above, to find the greatest real root of Eq. 3.

Theorem 1: The distribution function \( F \) does not affect the estimated value of \( r \) as computed in Eq. 3.
Proof: According to Eq. 3, the only way the value of \( r \) might depend on \( F \) is through \(-\ln(1 - F(x_i))\). However, we will show that for every \( i \), the value of \(-\ln(1 - F(x_i))\) does not depend on \( F \), namely, that the distribution of the random variable \( Y = -\ln(1 - F(X_i)) \) does not depend on \( F \) given \( r \) affected nodes.

Denote by \( f_{X_i|r}(x) \), for \( i = 1, \ldots, N \), the density function of \( X_i \) given that the number of affected nodes is \( r \). The function \( f_{X_i|r}(x) \) is actually the probability density of the \( r \)th order statistic of distribution \( F \), namely:

\[
f_{X_i|r}(x) = \frac{d}{dx} F_{X_i|r}(x) = \frac{1}{i} \int_0^x f(t) \left( 1 - F(t) \right)^{i-1} dt.
\]

Then, we have

\[
F_{X_i|r}(t) = \int_0^t f_{X_i|r}(x) dx = \frac{1}{i} \int_0^t \int_0^x f(t) \left( 1 - F(t) \right)^{i-1} dt dx.
\]

Substituting \( y = F(x) \), so that \( dy = f(x) dx = f(x) dx \), yields

\[
F_{X_i|r}(t) = \frac{1}{i} \int_0^t \int_0^x f(t) \left( 1 - F(t) \right)^{i-1} dt dy
\]

\[
= \frac{1}{i} \int_0^t (1 - F(t))^{i-1} y^{i-1} dy
\]

\[
= \frac{1}{i} \int_0^t (1 - F(t))^{i-1} \sum_{k=0}^{i-1} (-1)^k \binom{r - i}{k} y^k dy
\]

\[
= \frac{1}{i} \int_0^t (1 - F(t))^{i-1} \sum_{k=0}^{i-1} (-1)^k \binom{r - i}{k} F(t)^{k+i} dt.
\]

Let \( F_Y|z \) be the distribution function of \( Y \) given there are \( r \) affected nodes. Hence,

\[
F_Y|z = P(-\ln(1 - F(X_i)) \leq z | \text{ \# affected nodes})
\]

\[
= P(1 - F(X_i) \geq e^{-z} | \text{ \# affected nodes})
\]

\[
= P(F(X_i) \leq 1 - e^{-z} | \text{ \# affected nodes})
\]

\[
= P(X_i \leq F^{-1}(1 - e^{-z}) | \text{ \# affected nodes})
\]

\[
= F_{X_i|r}(F^{-1}(1 - e^{-z}))
\]

\[
= \frac{1}{i} \int_0^t (1 - F(t))^{i-1} \sum_{k=0}^{i-1} (-1)^k \binom{r - i}{k} (1 - e^{-z})^{k+i} dt.
\]

Thus, given \( r \) affected nodes, \( Y \) does not depend on \( F \).

In [19] it was shown that a truncated exponential distribution outperforms a uniform distribution at reducing feedback implosion when the time interval during which RPRTs are to be sent is limited. Theorem 1 suggests that this is not the case when setting a hard limit on the number of RPRTs. The time interval \([t_0, t_0 + T]\) during which the RPRTs are transmitted is arbitrary too, since changing \( T \) results in changing the support of the distribution.

We implemented the algorithm proposed in Section III for two distribution functions: the uniform distribution \( f(x) = \frac{1}{T} \) and the truncated exponential distribution \( f(x) = \frac{1}{x^e} \cdot \frac{1}{T} e^{-x} \), for \( x \in [0, T] \). In the simulation, the receivers draw timers from the above distributions and send a RPRT to the sender if fewer than \( N \) RPRTs have already been sent.

We have simulated 100, 1000 and 10,000 affected nodes. Figure 1 depicts the average error in estimating \( r \), namely \( |r_{\text{real}} - r_{\text{estimated}}|/r_{\text{real}} \), as a function of \( N \) for the uniform distribution and for the truncated exponential distribution. Each point in the graph is the average of 1000 different runs. This figure reveals that there is no noticeable difference between the uniform distribution and the truncated exponential distribution, as predicted by Theorem 1. It is evident that the greater \( N \) is, the better the estimation is.

Another important property of the proposed NEW! scheme is that it actually overestimates the number of affected nodes, i.e., with high probability \( r_{\text{estimated}} \geq r_{\text{real}} \). This is shown in Figure 2, which depicts the estimation error as \( (r_{\text{estimated}} - r_{\text{real}})/r_{\text{real}} \). We can see that for all values of \( N \) and for any number of affected nodes, the error is always positive. Moreover, by comparing the graphs in Figure 2 to those in Figure 1, we can see that the curves are almost identical. This implies that \( (r_{\text{estimated}} - r_{\text{real}})/r_{\text{real}} \approx (r_{\text{estimated}} - r_{\text{real}})/r_{\text{real}} \). We can, in other words, that the difference between the actual and the estimated values results from the overestimation.

We prove this property mathematically in Theorem 2 of Section IV, and show how it can be used in order to bring the estimation error very close to 0.

Implementation Notes:

1) RPRT messages are subject to loss due to collisions on a contention channel or to transmission errors. In Section V we show that NEW! can tolerate the loss of a small number of RPRTs, and that this number strongly depends on \( N \). For example, one loss when \( N = 4 \) and 5 losses when \( N = 100 \) have a small effect on the estimation error. To address the problem of message loss, we enhance NEW! with the following reliability mechanism. Upon receiving an RPRT, the gateway sends a confirmation to the sender. An RPRT sender that does not receive a confirmation within a time-out period resends its RPRT, and specifies the offset between its current local time and the time of the original RPRT.

2) In the protocol described so far, it was assumed that all the nodes are synchronized to a common clock or that all of them receive the START message almost at the same time. However, by requirement R5 from Section II, NEW! can also be executed in a system where there is no common clock and there is a variable delay between the gateway and the various nodes, provided that the gateway knows the delay \( D_i \) to/from every node \( i \). In such a system the following adjustment should be used. Let \( D_i \) be the delay from node \( i \) to the gateway, and let
\(D = \max_i (D_i)\) be the maximum delay. The gateway adds the value of \(D\) to the START message and each node \(i\) should wait a time period of \(D - D_i\) before running Algorithm 1.

3) To overcome a possible loss of the STOP message, the gateway must retransmit this message if it receives an RPRT after it sends a STOP, while taking into account the maximum round trip time from the gateway to the nodes.

4) When the delay from the gateway to the nodes is not negligible compared to the value of \(T\) from Algorithm 1, the gateway is likely to receive more RPRTs after sending a STOP message. This problem can be addressed by Algorithm 3 below.

**Algorithm 3:** The gateway algorithm when the delay to the gateway is not negligible compared to \(T\).

Let \(T_N(t)\) be the estimation made by the gateway at time \(t\) for the time elapsed until the \(N'\)th RPRT will be received. This estimation is based on the RPRTs received by the gateway until time \(t\). Let \(D\) be the propagation delay to the farthest node. When \(T_N(t) = D\), a STOP message has to be sent. Let \(N'\) be the number of RPRTs actually received by the gateway when it sends this STOP. If \(N' \neq N\), the gateway should use the value of \(N'\) rather than the value of \(N\) when solving Eq. 3.

The expected value of the time of the \(i\)th RPRT can be computed using Eq. 5 as follows:

\[
E(X_i) = \int_0^T x \cdot r \binom{r-1}{i-1} F(x)^{i-1} (1 - F(x))^{r-i} f(x) \, dx.
\]

This integral can be computed numerically for any distribution \(F\). For the uniform distribution it can also be computed analytically: substituting \(f(x) = 1\) and \(F(x) = x\) we get

\[
E(X_i) = r \binom{r-1}{i-1} \frac{i! (r-i)!}{(r+1)!} = \frac{i}{r+1}.
\]

Therefore, for a uniform distribution, if by time \(t\) the gateway receives \(n\) RPRTs, the \(N'\)th RPRT is expected to arrive at \(\frac{T}{r} \cdot N\), and \(T_N(t) = t \cdot \left(\frac{N}{r} - 1\right)\). It is evident from the graphs presented in Section IV that even in the unlikely event that \(N'\) is smaller than \(N\) by 20-30\%, the precision of the estimation is not affected at all.
IV. Precision Analysis and Error Cancellation

In this section we analyze the precision of our algorithm. The importance of this section is two-fold. First, we prove that the estimation error is approximately \( \frac{1}{r} \). Second, we use this analysis in order to further reduce the error, and to bring it very close to 0.

We have already shown that the timer distribution function \( F \) does not affect the result \( r \) of the estimation. Therefore, in the following analysis we consider a uniform timer distribution on the interval \([0, 1]\). On that interval, \( f(x) = 1 \) and \( F(x) = x \).

The gateway estimates the number of affected nodes by finding the maximal value of \( r \) that solves the following equation:

\[
1 \frac{1}{r} + 1 \frac{1}{r - 1} + \ldots + 1 \frac{1}{r - (N - 1)} = - \ln(1 - x_N). \tag{6}
\]

Let this solution be \( r = g(x_N) \). Let \( R \) be a random variable denoting the estimated number of affected nodes, and let its expected value be \( E(R) \). Our goal is to approximate the relative error \( \frac{E(R) - r}{r} \).

We have seen that

\[
f_{X|r}(x) = r \left( \frac{r - 1}{x - 1} \right) F(x)^{i-1} (1 - F(x))^{r-i} f(x). \tag{7}
\]

Since \( f(x) = 1 \) and \( F(x) = x \), then substituting \( i = N \) into Eq. 7, we get

\[
f_{X_N|r}(x) = r \left( \frac{r - 1}{N - 1} \right)^{N-1} (1-x)^{r-N}. \]

Therefore,

\[
E(R) = \int_0^1 g(x) f_{X_N|r}(x) \, dx. \tag{8}
\]

We used an iterative method in order to solve \( g(x) \). In what follows we seek for upper and lower bounds on this function. From Eq. 6 follows that \( - \frac{\ln(1 - x_N)}{x_N} \) and \( - \frac{N}{x_N} \). Therefore,

\[
- \frac{N}{\ln(1 - x_N)} \leq g(x_N) \leq - \frac{N}{\ln(1 - x_N)} + (N - 1). \tag{9}
\]

To find a lower bound we substitute the left-hand part of Eq. 9 into Eq. 8 and get

\[
E(R) \geq -rN \left( \frac{r - 1}{N - 1} \right) \int_0^1 \frac{x^{N-1}(1-x)^{r-N}}{\ln(1-x)} \, dx
\]

\[
= -rN \left( \frac{r - 1}{N - 1} \right) \int_0^1 \frac{x^{k-1}(1-x)^{N-1}}{\ln x} \, dx. \tag{10}
\]

Next, we note that

\[
\int_0^1 \frac{x^n(1-x)^k}{\ln x} \, dx = \sum_{i=0}^n (-1)^i \binom{k}{i} \ln(n+i+1), \tag{11}
\]

for \( n \geq 0, k \geq 1 \), and \( \int_0^1 \frac{x^n}{\ln x} \, dx = -\infty \) for \( n \geq 0, k = 0 \). This means that for \( N = 1 \), \( E(R) = \infty \) and the relative estimation error is also infinite. Thus, from now on we assume that \( N \geq 2 \). In the following equations, a sum or a product whose lower limit is greater than its upper limit is considered to be equal to 0 or to 1 respectively. Substituting Eq. 11 into Eq. 10, yields:

\[
E(R) \geq -rN \left( \frac{r - 1}{N - 1} \right) \sum_{i=0}^{N-1} (-1)^i \binom{N - 1}{i} \ln(r - (N - 1) + i) \tag{12}
\]

\[
= -rN \frac{N-2}{(N-1)!} \prod_{i=0}^{N-2} (r - (N - 1) + i) \tag{13}
\]

\[
\cdot \sum_{i=0}^{N-1} (-1)^i \binom{N - 1}{i} \ln(r - (N - 1) + i).
\]

We can expand the right item of this product as follows:

\[
\sum_{i=0}^{N-1} (-1)^i \binom{N - 1}{i} \ln(r - (N - 1) + i) = \ln(r - (N - 1)) + \sum_{i=1}^{N-1} (-1)^i \left( \binom{N - 2}{i - 1} + \binom{N - 2}{i} \right) \ln(r - (N - 1) + i) \]

\[
+ \sum_{i=1}^{N-1} (-1)^i \binom{N - 2}{i} \ln(r - (N - 1) + i) + (-1)^{N-1} \ln r \tag{14}
\]

We now group in Eq. 14 those \( \ln \) terms that are adjacent but differ in sign, and we get

\[
\ln(r - (N - 1)) - \sum_{i=0}^{N-3} (-1)^i \binom{N - 2}{i} \ln(r - (N - 1) + i) + \sum_{i=0}^{N-2} (-1)^i \binom{N - 2}{i} \ln(r - (N - 1) + i)
\]

\[
+ (-1)^{N-1} \ln r = \ln \frac{r - (N - 1)}{r - (N - 1) + i} + \sum_{i=1}^{N-3} (-1)^i \binom{N - 2}{i} \ln \frac{r - (N - 1) + i}{r - (N - 1) + i + 1} \]

\[
+ (-1)^{N-2} \ln \frac{r - 1}{r} = - \ln \left( 1 + \frac{1}{r - (N - 1)} \right) - \sum_{i=1}^{N-3} (-1)^i \binom{N - 2}{i} \ln \left( 1 + \frac{1}{r - (N - 1) + i} \right)
\]

\[
- (-1)^{N-2} \ln \left( 1 + \frac{1}{r - 1} \right) = - \sum_{i=0}^{N-2} (-1)^i \binom{N - 2}{i} \ln \left( 1 + \frac{1}{r - (N - 1) + i} \right). \]
where:

\begin{equation}
\frac{\prod_{i=0}^{N-2} \binom{N-2}{i}}{(N-1)!} \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \ln \left(1 + \frac{1}{r - (N - 1) + i}\right)
\end{equation}

**Proposition 2:** Let \( c_k \) be the coefficient of \( r^k \) in the polynomial \( \prod_{j \neq i}^{N-2} (r - (N - 1) + j) \), for \( 1 \leq k \leq N - 2 \). Then, there exists a polynomial \( p(x) \) of degree \( N - 2 - k \) such that \( c_k = p(i) \).

**Proof:** For \( 1 \leq k \leq N - 2 \),

\[
c_k = \sum_{A_1} \text{product of the factors } -[(N - 1) - j] = \sum_{A_2} \text{product of the factors } -[(N - 1) - j] + [(N - 1) - i].
\]

\[
\cdot \sum_{A_2} \text{product of the factors } -[(N - 1) - j] = a_k + [(N - 1) - i]c_{k+1},
\]

where:

- \( A_1 \) is the set of all ways to choose \( N - 2 - k \) \( j \)'s from \( \{0, \ldots, N - 2\}\) \( \backslash \{i\}\).
- \( A_2 \) is the set of all ways to choose \( N - 2 - k \) \( j \)'s from \( \{0, \ldots, N - 2\}\).
- \( A_3 \) is the set of all ways to choose \( N - 3 - k \) \( j \)'s from \( \{0, \ldots, N - 2\}\) \( \backslash \{i\}\).

- \( a_k \) is a constant with respect to \( i \).

For \( k = N - 2 \), \( c_{N-2} = 1 \). Therefore, the proposition holds for \( c_{N-2} \) with the constant polynomial \( p(x) = 1 \).

By reverse induction, suppose that the proposition holds for \( k + 1 \). Then, there is a polynomial \( q(x) \) of degree \( N - 3 - k \) such that \( c_{k+1} = q(i) \). Define \( p(x) = a_k + (N - 1)q(x) - xq(x) \). Then, \( p(x) \) is a polynomial of degree \( N - 2 - k \) and \( p(i) = a_k + (N - 1)q(i) - iq(i) = a_k + [(N - 1) - i]c_{k+1} = c_k \).

**Proposition 3:** For all \( n \geq 0 \), \( \sum_{i=0}^{n} \binom{n}{i} (-1)^i = 0 \). For all \( k \geq 1 \) and \( n \geq k + 1 \), \( \sum_{i=k}^{n} \binom{n}{i} (-1)^i(i-1) \ldots (i-(k-1)) = 0 \).

**Proof:** The first claim follows immediately from the binomial formula:

\[
\sum_{i=0}^{n} \binom{n}{i} (-1)^i = \sum_{i=0}^{n} \binom{n}{i} (-1)^i n^i = (1-1)^n = 0.
\]

The second claim is proved by differentiating \( (x-1)^n \) \( n \) times, first as a composite function and then after expansion using the binomial formula.

\[
[(x-1)^n]' = n[(x-1)^{n-1}]
\]

\[
[(x-1)^n]'' = n[(x-1)^{n-2}]
\]

\[
\vdots
\]

\[
[(x-1)^n]^{(k)} = n[(x-1)^{n-k}]
\]

\[
[(x-1)^n]^{(k)} = \left[ \sum_{i=0}^{n} \binom{n}{i} (-1)^i x^i \right]^{(k)} = (-1)^k \sum_{i=k}^{n} \binom{n}{i} (-1)^i \cdot i(i-1) \ldots (i-(k-1)) x^{i-k}.
\]

By substituting \( x = 1 \), the proof is completed.

**Proposition 4:** Every polynomial \( p(x) = \sum_{i=0}^{k} a_i x^i \) of degree \( k \) can be written as a linear combination of the polynomials in the set \( B_k = \{ 1, x, x(x-1), x(x-1)(x-2), \ldots, x(x-1) \ldots (x-(k-1)) \} \).

**Proof:** By induction on \( k \). For \( k = 0 \), \( p(x) = a_0 \) is certainly a linear combination of the polynomials in \( B_0 = \{ 1 \} \). For a general \( k \geq 1 \),

\[
p(x) = a_k \cdot x(x-1) \ldots (x-(k-1))
\]

\[
+ \{ p(x) - a_k \cdot x(x-1) \ldots (x-(k-1)) \}
\]

where the polynomial in brackets is of degree \( k - 1 \). Thus, by the induction hypothesis, it can be written as a linear combination of the polynomials in \( B_{k-1} \).

**Proposition 5:** For every \( k \geq 0 \) and \( n \geq k + 1 \),

\[
\sum_{i=0}^{n} \binom{n}{i} (-1)^i p_k(i) = 0,
\]

where \( p_k(x) \) is a polynomial of degree \( k \).

**Proof:** By Proposition 4 we can write

\[
p_k(i) = b + a_0 i + a_1 i(i-1) + \ldots + a_{k-1} i(i-1) \ldots (i-(k-1)).
\]

Then, by Proposition 3,

\[
\sum_{i=0}^{n} \binom{n}{i} (-1)^i p_k(i) = b \sum_{i=0}^{n} \binom{n}{i} (-1)^i + a_0 \sum_{i=1}^{n} \binom{n}{i} (-1)^i + \ldots
\]

\[
+ a_{k-1} \sum_{i=0}^{n} \binom{n}{i} (-1)^i(i-1) \ldots (i-(k-1)) = 0.
\]
We will now use the above propositions to simplify the expression for $E(R)$ from Eq. 16:

\[
\sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \left( \prod_{j \neq i} (r - (N - 1) + j) \right) = \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \left( \prod_{j \neq i} c_k r^k \right) = \sum_{k=1}^{N-2} r^k \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} c_k.
\]

By Proposition 2, $c_k$ can be written as a polynomial of degree $N - 2 - k$ for the variable $i$. Thus, by Proposition 5, the first term vanishes. Note that $c_0 = (-1)^{N-2} \frac{(N-1)!}{(N-1-k)!} = (-1)^N \frac{(N-1)!}{(N-1-k)!}$, and so the second term can be expanded as follows:

\[
r^0 \sum_{i=0}^{N-2} \binom{N-2}{i} (-1)^i c_0 = (-1)^N \sum_{i=0}^{N-2} \binom{N-2}{i} (-1)^i \frac{(N-1)!}{N - 1 - i} = (-1)^N (N-1)! \sum_{i=0}^{N-2} \binom{N-2}{i} (-1)^i \frac{1}{i + 1} = (N-1)! \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \frac{(N-2)!}{(i+1)!(N-2-i)!}.
\]

Denote $q = \ln \left( 1 + \frac{1}{r-1} \right)^{r-1}$. Then, $0 < q < 1$, and for large enough values of $r - 1$, $q$ is close to 1. Then,

\[
E(R) \leq q \cdot r \cdot \frac{N}{(N-1)!} \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \frac{N-1}{i+1} + (N-1) = q \cdot r \left( 1 + \frac{1}{N-1} \right) + (N-1).
\]

This completes the upper bound analysis. From Eq. 17 and Eq. 18 we conclude that:

\[
p \cdot r \left( 1 + \frac{1}{N-1} \right) \leq E(R) \leq q \cdot r \left( 1 + \frac{1}{N-1} \right) + (N-1)
\]

\[
p \left( 1 + \frac{1}{N-1} \right) \leq \frac{E(R)}{r} \leq q \left( 1 + \frac{1}{N-1} \right) + \frac{N-1}{r}
\]

\[
p \frac{N}{N-1} - (1-p) \leq \frac{E(R)-r}{r} \leq q \frac{N}{N-1} - (1-q) + \frac{N-1}{r}.
\]
Eq. 19 shows the upper and lower bounds on the estimation error as a function of the number of RPRTs N and the real number of affected nodes r.

**Theorem 2:** When $1 \ll N \ll r$, the estimation error is approximately $\frac{1}{N-1}$. Moreover, this error is positive, which means that our algorithm overestimates the number of affected nodes.

**Proof:** The proof follows from Eq. 19. When $1 \ll N \ll r$, $p$ and $q$ are close to 1, and we have $E(R) \approx r(1 + \frac{1}{N})$.

In Figure 3 we show again the estimation error as obtained from simulations for 10,000 affected nodes. However, this time we compare this error to our analytic prediction $f(N) = \frac{1}{N-1}$ and see excellent agreement between the two curves.

Using the above analysis, we now show how to eliminate the estimation error with no further cost. Let $r_{\text{real}}$ be the real number of affected nodes and $r_{\text{estimate}}$ be the estimated number using our estimation algorithm. From the above analysis, we know that

$$r_{\text{real}} \approx r_{\text{estimate}} \cdot \frac{N - 1}{N}. \quad (20)$$

Figure 4 shows simulation results of this process for 1,000 and 10,000 affected nodes before and after applying “error cancellation”. It is evident that after applying error cancellation our algorithm reduces the error to approximately 0 even when the number of RPRTs is very small (3-4).

V. THE EFFECT OF RPRT LOSS

In the analysis so far, we have assumed that RPRTs are not lost. We have also presented, in the Implementation Notes of Section III, a simple mechanism for the retransmission of lost RPRTs by their senders. In this section we study the effect of RPRT losses on NATO! and show why such a retransmission mechanism is indeed necessary.

Suppose that by the time the gateway receives the $N$th RPRT, $S$ RPRTs have been lost. Therefore, the RPRT considered by the gateway to be the $N$th is actually the $N + S$th.

This influences Eq. 6, which the gateway solves to find $r$. This equation now should read:

$$\frac{1}{r} + \frac{1}{r-1} + \ldots + \frac{1}{r-(N-1)} = -\ln(1-x_{N+S}). \quad (21)$$

Going through the analysis in Section IV and replacing occurrences of $N$ with $N + S$ when $N$ indicates the index of the $N$th RPRT (as opposed to places where it indicates the number of RPRTs), we get:

$$-p \frac{S - 1}{N + S - 1} - (1 - p) \leq \frac{E(R) - r}{r} \leq -q \frac{S - 1}{N + S - 1} - (1 - q) + \frac{N - 1}{r}. \quad (22)$$

In this equation, $p = \ln \left(1 + \frac{1}{r(N+S)}\right)^{r-(N+S-1)}$ and $q = \ln \left(1 + \frac{1}{r}\right)^{r-1}$. This time, when $r \gg N$ and $N + S \gg 1$, the estimation error is approximately

$$\frac{E(R) - r}{r} \approx -\frac{S - 1}{N + S - 1}. \quad (23)$$

Figure 5 shows the estimation error in the presence of RPRT loss. We consider 1 or 5 RPRT losses, and show the error using both simulation and Eq. 23. A few conclusions can be drawn from this figure. First, as in Figure 3, we can see very good agreement between the simulation and the analysis. Second, comparing Figure 5(a) to Figure 5(b) reveals no major difference. This implies that, as before, the error is independent of $r$. Third, as expected (see Eq. 23), the error now is negative. Thus, in the presence of RPRT loss, our method underestimated the number of affected nodes instead of overestimating it. However, the most important conclusion we draw from the figure is that when RPRTs are subject to loss, NATO! should be enhanced with a mechanism for RPRT retransmissions, as proposed in the Implementation Notes.

An interesting observation is that when $S = 1$, the error almost vanishes. This is also implied by Eq. 23, which predicts a 0 error in this situation. This fact has a rather simple explanation. As noted above, RPRT losses cause our method to underestimate the error, while with no RPRT losses the error is overestimated. When $S = 1$, the two counter-forces cancel each other, resulting in an error very close to 0.

We can apply the same process used in the end of Section IV to reduce the effect of RPRT loss on the precision of our algorithm. In this case, Eq. 20 is replaced by the following equation:

$$r_{\text{real}} \approx r_{\text{estimate}} \cdot \frac{N + S - 1}{N}. \quad (24)$$

Figure 6 presents the error for $S = 10$ lost RPRTs before and after applying the error cancellation process. Note, however, that in order to implement this process when $S > 0$, the gateway needs to know the value of $S$, which is not always possible.
VI. CONCLUSIONS AND FUTURE WORK

This paper is the first to explicitly show the correlation between the number of RPRTs sent by a group of affected nodes and the ability of the gateway to precisely estimate the size of this group. We developed a statistical analysis algorithm for estimating the number of affected nodes. The algorithm, which is based on the times the RPRTs are received, defines the likelihood function for the received RPRTs and then uses the Newton-Raphson method to find the number of bad receivers for which this function is maximized. We analyzed the error of our algorithm and showed that when $1 \ll N \ll r$, where $r$ is the number of affected nodes and $N$ is the number of RPRTs, this error is positive and approximately equal to $1/(N-1)$. We used this important result to correct the estimation of our algorithm, and to bring it very close to 0.

NATO! is an important building block in several important application scenarios, where thousands of nodes need to send report messages to a common gateway. However, the integration of NATO! into the various application scenarios has not been addressed in this paper. In [9] we address this issue in the context of reliable multicast in wireless/cellular/satellite networks. However, more work has to be done in the scope of other important applications, such as sensor networks and grid networks. For example, in sensor networks it is important to determine the value of $T$ as a function of the emergence of the event detected by the sensors and of the propagation time from the most distant sensor to the gateway.

REFERENCES

Fig. 6. Estimation error vs. $N$ for $S = 10$, before and after “error cancellation”

(a) $r = 1,000$ affected nodes

(b) $r = 10,000$ affected nodes


