A Generic Scheme for Estimating the Size of a Group While Avoiding Feedback Implosion

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

***Abstract*—We present a probabilistic polling scheme for estimating the size of a group of nodes affected by the same event. We analyze the bias of the proposed scheme and show how it can be completely eliminated. Our scheme differs from previous work in several important ways. First, it is generic in the sense that it is not dependent on the physical properties of the underlying network. Second, it uses a "one-shot" estimation technique that does not depend on the results of previous rounds. Finally, the estimating nodes control the number of feedback messages, thus allowing a good balance between the overhead imposed by the scheme and its precision. We compared our scheme to previous works and saw a notable performance improvement: our scheme reduced the number of response messages by 80-90% while obtaining the same precision.**

***Index Terms*—feedback implosion, group size estimation**

I. Introduction

We propose a new probabilistic polling scheme for estimating the size of a group of nodes that experience some event. We call this scheme *NATO! (Not All aT Once!)*. When the estimating node, also referred to as the "gateway," announces that the polling will begin, every group member chooses a random number from the interval $[0, 1]$ using a known distribution. The N smallest random numbers are then collected by the gateway. Our new maximum likelihood algorithm uses the Newton-Raphson method to estimate the number of affected nodes using these N smallest random numbers. We provide tight upper and lower bounds for the bias of the proposed algorithm and use these bounds in order to remove it.

While the proposed scheme is generic, we show how it can be optimized for networks with a broadcast channel, such as satellite networks, sensor networks, and broadband access wireless networks. In such networks, the gateway collects the response messages by means of an efficient distributed protocol, based on the well-known Capetanakis algorithm [8].

Probabilistic polling for estimating the size of a group has been studied in several papers [3], [4], [6], [10], [11], [12], [13]. These papers are discussed in greater detail in Section II. Our work differs from them in four important ways:

(A1) The proposed algorithm is generic in the sense that its precision does not depend on the properties of the underlying network. In other words, we decouple these properties from the obtained precision.

(A2) The proposed algorithm uses a "one-shot" estimation technique that does not depend on the results of previous rounds. Therefore, it can detect events for which the population size changes rapidly, such as a denial of service attack.

(A3) In our scheme, the group estimation problem is viewed as a generic *cost vs. profit* problem. The cost is the number of feedback messages received from the group and the profit is the precision of the estimation. The estimating node can determine how many feedback messages it wants to receive if it knows what error it can tolerate. Our work is probably the first to show the number of feedback messages needed to obtain a precise estimation within a certain confidence interval.

(A4) The proposed algorithm can be implemented in a network with a shared channel such that losses, either due to collisions or to transmission errors, are overcome. This is possible because the algorithm does not depend on the times when the response messages are sent.

But NATO! is an important contribution to the field of probabilistic polling not only because it is unique in the four aspects discussed above, but also because it performs very well in comparison to previous algorithms. Comparing the performance of probabilistic polling algorithms is difficult, and sometimes not really "fair," because different algorithms make different assumptions and often seek to optimize different parameters, as discussed in Section II. Moreover, not all previous works present their cost (number of feedback messages) vs. accuracy results. But by comparing our results to those that have been published, we see a major improvement. For example, in [10] the authors indicate that 423 messages are required to estimate the size of a group at 95% confidence interval and and error smaller than 10%. In [11], it is shown that the "NB scheme" needs 230 messages, one during each round, for similar accuracy. In Section IV of this paper, we show that less than 30 messages are required by our NATO! scheme.

There are many applications for the proposed scheme, a few of which we elaborate on here. It can be applied, for instance, to detect denial of service jamming attacks in broadcast wireless networks [18]. Using the proposed scheme, the base station can periodically estimate the number of nodes that are able to receive its broadcast messages without requiring each of them to send an individual response. If the estimated group is much smaller than the number of registered nodes, the base station can deduce that its broadcast channel is being jammed. For this application, NATO! has to be executed in a contention wireless channel. A possible implementation is presented in Section V.

The scheme can also be used in reliable multicast [9]. 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], [14], [16], receivers that have not received at least K packets correctly notify the sender, by means of a NACK message, and the sender transmits additional repair packets. The number of repair rounds is usually limited by real-time considerations. In reliable multicast, NATO! can be invoked by the sender once every time out in order to estimate the loss distribution for the considered multicast group. That is, the sender will estimate the number of nodes in this group that have lost $p\%$ of multicast packets since the previous round, for several relevant values of p. Using this information, the sender can then 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.

Another application for NATO! is feedback implosion in sensor networks. Consider sensor networks where the gateway periodically asks the nodes to report about specific events, such as temperature exceeding some threshold. Most papers that address this problem adopt the concept of data aggregation, where similar data messages sent by multiple sources are aggregated by the network nodes. However, in single hop sensor networks, where all the sensors have a wireless connection to the gateway, the NATO! scheme allows the gateway to determine the number of report messages it wants to receive for every event.

The rest of this paper is organized as follows. In Section II we present related work. In Section III we present the estimation algorithm. In Section IV we analyze the precision of this algorithm and find tight upper and lower bounds for the bias. Using this analysis, we bring the bias of our estimator to 0. In this section we also show the trade-off between the number of response messages and the precision of the new algorithm. In Section V, we first show that the loss of response messages has a negative impact on the performance of the estimation algorithm. Then, we show how losses can be overcome in networks with a broadcast channel, such as broadband access wireless networks. Finally, Section VI concludes the paper.

II. Related Work

The authors of [5] are probably the first to consider the problem of estimating the size of a group in a network using polling protocols. They also defined several cost functions for this problem. In [6], a multi-round scheme, also known as the “BTW scheme,” is proposed in the context of multicast flow control. For each round n, the reply probability p_n is defined by the estimating node in the request for feedback that it sends. If no reply is received, the estimating node increases the reply probability for the next round until a response is received. The population size is then estimated using this single response.

Nonnenmacher and Biersack were probably the first to analyze timer-based schemes [12], [13]. They proposed the “NB scheme,” which also uses multiple rounds. For each round i, the estimating node sends a polling message *RFB*(i) with the distribution $F^{(i)}(z)$, $z \in (0, T)$. Each receiver j then draws a time $z_j^{(i)}$ from $F^{(i)}(z)$ and sends a response message after this time if the round is not terminated by a new probing message *RFB*$(i + 1)$. The estimating node starts a new round after receiving the first response message. The delay between the estimating node and the receivers is assumed to be constant.

In [10], Friedman and Towsley view the group estimation problem as estimating the parameter N of the binomial (N, p) distribution. In each round i, the estimating node multicasts a polling request. A receiver sends a response message with probability p_i. After k rounds, the sender estimates the value of n from the polling probabilities $p_1, p_2, \ldots, p_k$ and from the number of responses $r_1, r_2, \ldots, r_k$ in every round. This paper also shows how to map the BTW scheme and the timer-based NB scheme into its binomial estimation model. In addition, it defines a maximum likelihood estimator for the NB scheme that uses information from multiple rounds.

The work that most resembles our is probably [11] in that it also uses a maximum likelihood algorithm that takes into account several feedback messages. When the first feedback message arrives at the gateway, it broadcasts the next RFB. The next RFB starts a new polling round and stops the receivers from sending additional responses to the previous round. The number of feedback messages received by the sender is therefore proportional to the length of the RTT.

Despite of the similarity to our work, [11] is different by all aspects (A1)-(A4) discussed in Section I. In our scheme, the number of feedback messages received by the gateway is determined by the gateway and not by the properties of the network. We overcome feedback implosion by limiting the number of response messages sent by the receivers, while [11] limits the time during which these messages are sent. In broadband access networks where the RTT is small and homogeneous, the estimating node in [11] is likely to receive only one feedback message for every RFB, in which case the precision is very limited. In contrast, a gateway that runs NATO! terminates the polling after it receives the required number of feedback messages to obtain the necessary precision. Our scheme also differs from [11] in that it requires only a single polling round, and we explicitly show the trade-off between the number of feedback messages and the precision of the algorithm. Finally, our scheme can tolerate retransmissions of response messages due to collisions or transmission errors.

In [3], [4], [2], an $M/M/\infty$ model for receivers entering and exiting the multicast group is discussed. 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 [7] consider a network with thousands of nodes connected to the same wireless channel and a gateway that needs to estimate their total number. However, the gateway does not use a probabilistic polling algorithm. Rather, it observes the regular traffic in the network and compares the identity of the sender of every message to the identities of the senders in previous messages. The authors compare a Good-Turing estimator to a maximum likelihood estimator and show that both have similar performance. Since this paper does not use specific messages, and does not limit the nodes to send at most one message per node, many more messages are required than in our NATO! scheme. While our scheme requires a small constant number of messages (≈ 40) to estimate a large population of 10,000 nodes with a good precision, the scheme in [7] requires $O\sqrt{r}\log r)$ messages, where r is the group size.

III. The Estimation Algorithm of NATO!

Let r be the number of affected nodes (i.e., estimated group size). The gateway announces the beginning of an estimation process by broadcasting a START message. After it receives N responses, it broadcasts a STOP message. When an affected node receives a START messages, it executes the following algorithm:

- Choose a random number t in the range $[0,1]$ using a known probability distribution function F.
- Send a RPRT(t) message to the gateway after all the nodes whose random numbers are smaller than t have sent their RPRTs, provided that a STOP message has not been received in the meantime. ■

How to guarantee that the nodes with the smallest random numbers will be the first to send their RPRT messages is an implementation detail *that is orthogonal to the NATO! scheme*, because different networks will have different implementations. Of course, it is possible to use a timer that is proportional to the random number drawn by every node, as in the timer-based polling schemes discussed in Section II. However, any scheme must guarantee that no RPRT messages will be lost because lost messages will seriously reduce the algorithm's precision. In Section V we present a possible implementation for networks with a contention channel.

Let f be the probability density function of F. Let $X_1, \ldots, X_N$ be random variables denoting the N smallest random numbers chosen by the affected nodes. 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.

We use the maximum likelihood method to estimate r. Let $f_{X_1,X_2,\ldots,X_N|r}(x_1, x_2, \ldots, x_N)$ 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 [15]:

$$\begin{aligned}
&f_{X_1,X_2,\ldots,X_r}(x_1, x_2, \ldots, x_r)dx_1 \ldots dx_r = \\
&\quad = P(X_1 \in (x_1, x_1 + dx_1), \ldots, X_r \in (x_r, x_r + dx_r)) = \\
&\quad = r! P(Y_1 \in (x_1, x_1 + dx_1), \ldots, Y_r \in (x_r, x_r + dx_r)) = \\
&\quad = r!(F(x_1 + dx_1) - F(x_1)) \ldots (F(x_r + dx_r) - F(x_r)) = \\
&\quad = r! f(x_1) \ldots f(x_r)\, dx_1 \ldots dx_r,
\end{aligned}$$

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 $x_{N+1}, \ldots, x_r$:

$$\begin{aligned}
&f_{X_1,X_2,\ldots,X_N|r}(x_1, x_2, \ldots, x_N) \\
&\quad = \int \ldots \int_{x_N < x_{N+1} < \ldots < x_r} f_{X_1,X_2,\ldots,X_r}(x_1, x_2, \ldots, x_r)\, dx_{N+1} \ldots dx_r \\
&\quad = \int \ldots \int_{x_N < x_{N+1} < \ldots < x_r} r! f(x_1) \ldots f(x_r)\, dx_{N+1} \ldots dx_r \\
&\quad = r! \int_{x_N} dx_{N+1} \ldots \int_{x_{r-1}} dx_r f(x_1) \ldots f(x_r) \\
&\quad = r! \int_{x_N} dx_{N+1} \ldots \int_{x_{r-2}} dx_{r-1} \prod_{i=1}^{r-1} f(x_i) \cdot (1 - F(x_{r-1})) \\
&\quad = r! \int_{x_N} dx_{N+1} \ldots \int_{x_{r-3}} dx_{r-2} \prod_{i=1}^{r-2} f(x_i) \cdot \frac{(1 - F(x_{r-2}))^2}{2} \\
&\quad = \ldots = \frac{r!}{(r-N)!} \prod_{i=1}^{N} f(x_i) \cdot (1 - F(x_N))^{r-N}. \qquad (1)
\end{aligned}$$

Define the likelihood function $L(r)$ to be

$$L(r) = f_{X_1,X_2,\ldots,X_N|r}(x_1, x_2, \ldots, x_N).$$

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

$$\begin{aligned}
l(r) &= \ln L(r) \\
&= \ln \frac{r!}{(r-N)!} + \ln f(x_1) + \ldots + \ln f(x_N) \\
&\quad + (r-N)\ln(1 - F(x_N)) \\
&= \ln(r - N + 1) + \ldots + \ln r \\
&\quad + r\ln(1 - F(x_N)) + const. \qquad (2)
\end{aligned}$$

In this equation, *const* is a constant with respect to r.

We now differentiate $l(r)$ with respect to r and get

$$l'(r) = \tfrac{1}{r} + \tfrac{1}{r-1} + \ldots + \tfrac{1}{r-N+1} + \ln(1 - F(x_N)).$$

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:

$$\frac{1}{r} + \frac{1}{r-1} + \ldots + \frac{1}{r-N+1} + \ln(1 - F(x_N)) = 0. \qquad (3)$$

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

$$l''(r) = -\frac{1}{r^2} - \frac{1}{(r-1)^2} - \ldots - \frac{1}{(r-N+1)^2} < 0, \qquad (4)$$

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:

$$\begin{aligned} l(r^*) &= \ln r^* + \ldots + \ln(r^* - N + 1) \\ &\quad -r^*\left(\frac{1}{r^*} + \frac{1}{r^*-1} + \ldots + \frac{1}{r^*-N+1}\right) + const \\ &= \ln r^* + \ldots + \ln(r^* - N + 1) - 1 \\ &\quad -\left(1 + \frac{1}{r^*-1}\right) - \ldots - \left(1 + \frac{N-1}{r^*-N+1}\right) + const. \end{aligned}$$

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 $\frac{1}{r} + \frac{1}{r-1} + \ldots + \frac{1}{r-N+1} + c = 0$. 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 ε 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 monotonic at the interval $(N-1, \infty)$, it is implied that $x_0 < root$. 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 (nth) 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 1: The gateway algorithm.

- Broadcast/multicast a START message to all possible affected nodes.
- When N RPRTs messages 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 absolutely continuous 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 $-\ln(1-F(x_N))$. However, we will show now 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 ith order statistic of distribution F, namely:

$$\begin{aligned} f_{X_i|r}(x) &= \frac{d}{dx}F_{X_i|r}(x) = \frac{d}{dx}P(X_i \leq x|r \text{ affected nodes}) \\ &= \frac{d}{dx}P(\text{at least } i \text{ of the } r \text{ random numbers are } < x) \\ &= \frac{d}{dx}\sum_{j=i}^{r}\binom{r}{j}F(x)^j(1-F(x))^{r-j} = \ldots \\ &= r\binom{r-1}{i-1}F(x)^{i-1}(1-F(x))^{r-i}f(x). \end{aligned}$$

Then, we have

$$\begin{aligned} &F_{X_i|r}(t) = \\ &\quad = \int_0^t r\binom{r-1}{i-1}F(x)^{i-1}(1-F(x))^{r-i}f(x)\,dx. \quad (5) \end{aligned}$$

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

$$\begin{aligned} F_{X_i|r}(t) &= r\binom{r-1}{i-1}\int_0^{F(t)} y^{i-1}(1-y)^{r-i}\,dy \\ &= r\binom{r-1}{i-1}\int_0^{F(t)} y^{i-1}\sum_{k=0}^{r-i}(-1)^k\binom{r-i}{k}y^k\,dy \\ &= r\binom{r-1}{i-1}\sum_{k=0}^{r-i}(-1)^k\binom{r-i}{k}\int_0^{F(t)} y^{k+i-1}\,dy \\ &= r\binom{r-1}{i-1}\sum_{k=0}^{r-i}(-1)^k\binom{r-i}{k}\frac{F(t)^{k+i}}{k+i}. \end{aligned}$$

Let $F_{Y|r}(z)$ be the distribution function of Y given there are r affected nodes. Hence,

$$\begin{aligned} F_{Y|r}(z) &= P(-\ln(1-F(X_i)) \leq z|r \text{ affected nodes}) \\ &= P(1-F(X_i) \geq e^{-z}|r \text{ affected nodes}) \\ &= P(F(X_i) \leq 1-e^{-z}|r \text{ affected nodes}) \\ &= P(X_i \leq F^{-1}(1-e^{-z})|r \text{ affected nodes}) \\ &= F_{X_i|r}(F^{-1}(1-e^{-z})) \\ &= r\binom{r-1}{i-1}\sum_{k=0}^{r-i}(-1)^k\binom{r-i}{k}\frac{(1-e^{-z})^{k+i}}{k+i}. \end{aligned}$$

Thus, given r affected nodes, Y does not depend on F. ■

IV. Precision Analysis and Bias Removal

In this section we analyze the accuracy of our algorithm. We prove that the bias is approximately $\frac{1}{N-1}$, and use this result in order to remove it.

We have already shown that the distribution function F does not affect the result r of the estimation. Therefore, in the following analysis we consider a uniform 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:

$$\frac{1}{r}+\frac{1}{r-1}+\ldots+\frac{1}{r-(N-1)}=-\ln(1-x_N). \qquad (6)$$

Let this solution be $r=g(x_N)$. Let $\widehat{r_1}$ be our estimator, i.e., a random variable denoting the estimated number of affected nodes, and let its expected value be $E(\widehat{r_1})$. Our goal is to approximate the bias $\frac{E(\widehat{r_1})-r}{r}$.

We have seen that

$$f_{X_i|r}(x)=r\binom{r-1}{i-1}F(x)^{i-1}(1-F(x))^{r-i}f(x). \qquad (7)$$

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

$$f_{X_N|r}(x)=r\binom{r-1}{N-1}x^{N-1}(1-x)^{r-N}.$$

Therefore,

$$E(\widehat{r_1})=\int_0^1 g(x)f_{X_N|r}(x)\,dx. \qquad (8)$$

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 $-\ln(1-x_N)\geq\frac{N}{r}$ and $-\ln(1-x_N)\leq\frac{N}{r-(N-1)}$. Therefore,

$$-\frac{N}{\ln(1-x_N)}\leq g(x_N)\leq-\frac{N}{\ln(1-x_N)}+(N-1). \qquad (9)$$

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

$$\begin{aligned}E(\widehat{r_1})&\geq -rN\binom{r-1}{N-1}\int_0^1\frac{x^{N-1}(1-x)^{r-N}}{\ln(1-x)}\,dx\\&= -rN\binom{r-1}{N-1}\int_0^1\frac{x^{r-N}(1-x)^{N-1}}{\ln x}\,dx. \qquad (10)\end{aligned}$$

Next, we note that

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

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 $N=1$, $E(\widehat{r_1})=\infty$ and the bias 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:

$$\begin{aligned}E(\widehat{r_1})&\geq -rN\binom{r-1}{N-1} \qquad (12)\\&\cdot\sum_{i=0}^{N-1}(-1)^i\binom{N-1}{i}\ln(r-(N-1)+i)\\&= -\frac{rN}{(N-1)!}\prod_{i=0}^{N-2}(r-(N-1)+i) \qquad (13)\\&\cdot\sum_{i=0}^{N-1}(-1)^i\binom{N-1}{i}\ln(r-(N-1)+i).\end{aligned}$$

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

$$\begin{aligned}&\sum_{i=0}^{N-1}(-1)^i\binom{N-1}{i}\ln(r-(N-1)+i)\\&= \ln(r-(N-1))\\&+\sum_{i=1}^{N-2}(-1)^i\left[\binom{N-2}{i-1}+\binom{N-2}{i}\right]\\&\cdot\ln(r-(N-1)+i)+(-1)^{N-1}\ln r\\&= \ln(r-(N-1))\\&+\sum_{i=0}^{N-3}(-1)^{i+1}\binom{N-2}{i}\ln(r-(N-1)+i{+}1)\\&+\sum_{i=1}^{N-2}(-1)^i\binom{N-2}{i}\ln(r-(N-1)+i)\\&+(-1)^{N-1}\ln r. \qquad (14)\end{aligned}$$

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

$$\begin{aligned}&\ln(r-(N-1))\\&-\sum_{i=0}^{N-3}(-1)^i\binom{N-2}{i}\ln(r-(N-1)+i+1)\\&+\sum_{i=1}^{N-2}(-1)^i\binom{N-2}{i}\ln(r-(N-1)+i)\\&+(-1)^{N-1}\ln r=\\&= \ln\frac{r-(N-1)}{r-(N-1)+1}+\\&+\sum_{i=1}^{N-3}(-1)^i\binom{N-2}{i}\ln\frac{r-(N-1)+i}{r-(N-1)+i+1}+\\&+(-1)^{N-2}\ln\frac{r-1}{r}=\\&= -\ln\left(1+\frac{1}{r-(N-1)}\right)-\\&-\sum_{i=1}^{N-3}(-1)^i\binom{N-2}{i}\ln\left(1+\frac{1}{r-(N-1)+i}\right)-\\&-(-1)^{N-2}\ln\left(1+\frac{1}{r-1}\right)=\\&= -\sum_{i=0}^{N-2}(-1)^i\binom{N-2}{i}\ln\left(1+\frac{1}{r-(N-1)+i}\right).\end{aligned}$$

Substituting this into Eq. 13 yields

$$\begin{aligned} E(\widehat{r_1}) \geq \frac{rN}{(N-1)!} \prod_{i=0}^{N-2} (r-(N-1)+i) \qquad (15) \\ \cdot \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \ln\left(1+\frac{1}{r-(N-1)+i}\right) \\ = \frac{rN}{(N-1)!} \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \\ \cdot \left[\prod_{\substack{j=0\\ j\neq i}}^{N-2} (r-(N-1)+j)\right] \\ \cdot \ln\left(1+\frac{1}{r-(N-1)+i}\right)^{r-(N-1)+i} . \end{aligned}$$

The sequence

$$\left\{\left(1+\frac{1}{r-(N-1)+i}\right)^{r-(N-1)+i}\right\}_{i=0}^{N-2}$$

is monotonically increasing and upper bounded by e. Denote

$$p = \ln\left(1+\frac{1}{r-(N-1)}\right)^{r-(N-1)} .$$

Then, $0 < p < 1$, and for large enough values of $r-(N-1)$, p is close to 1. We now get:

$$\begin{aligned} E(\widehat{r_1}) \geq \frac{p \cdot r \cdot N}{(N-1)!} \qquad (16) \\ \cdot \sum_{i=0}^{N-2} (-1)^i \binom{N-2}{i} \left[\prod_{\substack{j=0\\ j\neq i}}^{N-2} (r-(N-1)+j)\right] . \end{aligned}$$

We continue with a series of propositions that will help us to simplify the above expression.

Proposition 2: Let c_k be the coefficient of r^k in the polynomial $\prod_{\substack{j=0\\ j\neq i}}^{N-2}(r-(N-1)+j)$, 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 $1 \leq k \leq N-2$,

$$\begin{aligned} c_k = \sum_{A_1} \text{product of the factors } -[(N-1)-j] = \\ = \sum_{A_2} \text{product of the factors } -[(N-1)-j] + \\ +[(N-1)-i] \cdot \\ \cdot \sum_{A_3} \text{product of the factors } -[(N-1)-j] = \\ = a_k + [(N-1)-i]c_{k+1}, \end{aligned}$$

where:

- A_1 is the set of all ways to choose $N-2-k$ j's from $\{0,\dots,N-2\}\backslash\{i\}$.
- A_2 is the set of all ways to choose $N-2-k$ j's from $\{0,\dots,N-2\}$.
- A_3 is the set of all ways to choose $N-3-k$ j's from $\{0,\dots,N-2\}\backslash\{i\}$.
- 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 $n \geq k+1$, $\sum_{i=k}^{n} \binom{n}{i}(-1)^i i(i-1)\dots(i-(k-1)) = 0$.

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

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

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

$$\begin{aligned} [(x-1)^n]^{(k)} &= n(n-1)\dots(n-(k-1))(x-1)^{n-k} \\ [(x-1)^n]^{(k)} &= \left[\sum_{i=0}^{n} \binom{n}{i}(-1)^{n-i}x^i\right]^{(k)} = \\ &= (-1)^n \sum_{i=k}^{n} \binom{n}{i}(-1)^i \cdot \\ &\quad \cdot i(i-1)\dots(i-(k-1))x^{i-k}. \end{aligned}$$

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 $\mathcal{B}_k = \{1, x, x(x-1), x(x-1)(x-2), \dots, x(x-1)\dots(x-(k-1))\}$.

Proof: By induction on k. For $k = 0$, $p(x) = a_0$ is certainly a linear combination of the polynomials in $\mathcal{B}_0 = \{1\}$. For a general $k \geq 1$,

$$\begin{aligned} p(x) &= a_k \cdot x(x-1)\dots(x-(k-1)) \\ &\quad + [p(x) - a_k \cdot x(x-1)\dots(x-(k-1))] , \end{aligned}$$

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 $\mathcal{B}_{k-1}$. ■

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

$$p_k(i) = b + a_0 i + a_1 i(i-1) + \dots + a_{k-1} i(i-1)\dots(i-(k-1)).$$

Then, by Proposition 3,

$$\begin{aligned} &\sum_{i=0}^{n} \binom{n}{i}(-1)^i p_k(i) \\ &\quad = b\sum_{i=0}^{n} \binom{n}{i}(-1)^i + a_0 \sum_{i=1}^{n} \binom{n}{i}(-1)^i i + \dots \\ &\quad\quad + a_{k-1} \sum_{i=k}^{n} \binom{n}{i}(-1)^i i(i-1)\dots(i-(k-1)) \\ &\quad = 0. \end{aligned}$$

$\blacksquare$

We will now use the above propositions to simplify the expression for $E(\widehat{r_1})$ from Eq. 16:

$$\begin{aligned}
&\sum_{i=0}^{N-2}(-1)^i\binom{N-2}{i}\left[\prod_{\substack{j=0\\ j\neq i}}^{N-2}(r-(N-1)+j)\right]\\
&= \sum_{i=0}^{N-2}(-1)^i\binom{N-2}{i}\left[\sum_{k=0}^{N-2}c_k r^k\right]\\
&= \sum_{k=1}^{N-2} r^k\left[\sum_{i=0}^{N-2}\binom{N-2}{i}(-1)^i c_k\right]\\
&\quad +r^0\sum_{i=0}^{N-2}\binom{N-2}{i}(-1)^i c_0.
\end{aligned}$$

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 $c_0 = (-1)^{N-2}\frac{(N-1)!}{(N-1)-i} = (-1)^N\frac{(N-1)!}{(N-1)-i}$, and so the second term can be expanded as follows:

$$\begin{aligned}
&r^0\sum_{i=0}^{N-2}\binom{N-2}{i}(-1)^i c_0\\
&= (-1)^N\sum_{i=0}^{N-2}\binom{N-2}{i}(-1)^i\frac{(N-1)!}{N-1-i}\\
&= (-1)^N(N-1)!\sum_{i=0}^{N-2}\binom{N-2}{i}(-1)^{N-i}\frac{1}{i+1}\\
&= (N-1)!\sum_{i=0}^{N-2}\binom{N-2}{i}(-1)^i\frac{1}{i+1}\\
&= (N-1)!\sum_{i=0}^{N-2}(-1)^i\frac{(N-2)!}{(i+1)!(N-2-i)!}\\
&= (N-1)!\frac{1}{N-1}\sum_{i=0}^{N-2}(-1)^i\frac{(N-1)!}{(i+1)!(N-2-i)!}\\
&= (N-2)!\sum_{i=0}^{N-2}(-1)^i\binom{N-1}{i+1}\\
&= (N-2)!\sum_{i=1}^{N-1}(-1)^{i-1}\binom{N-1}{i}\\
&= -(N-2)!\sum_{i=1}^{N-1}(-1)^i\binom{N-1}{i}\\
&= -(N-2)!\left(\sum_{i=0}^{N-1}(-1)^i\binom{N-1}{i}-1\right)\\
&= (N-2)!,
\end{aligned}$$

where the last equality follows from Proposition 3.

Therefore, from Eq. 16, it follows that

$$\begin{aligned}
E(\widehat{r_1}) &\geq \frac{p\cdot r\cdot N}{(N-1)!}(N-2)! = p\cdot r\frac{N}{N-1}\\
&= p\cdot r\left(1+\frac{1}{N-1}\right). \qquad (17)
\end{aligned}$$

This completes the lower bound analysis for $E(\widehat{r_1})$.

To find an upper bound for $E(\widehat{r_1})$, we employ similar techniques. In the following equations, a number above a relation symbol denotes the number of an equivalent equation for the lower bound:

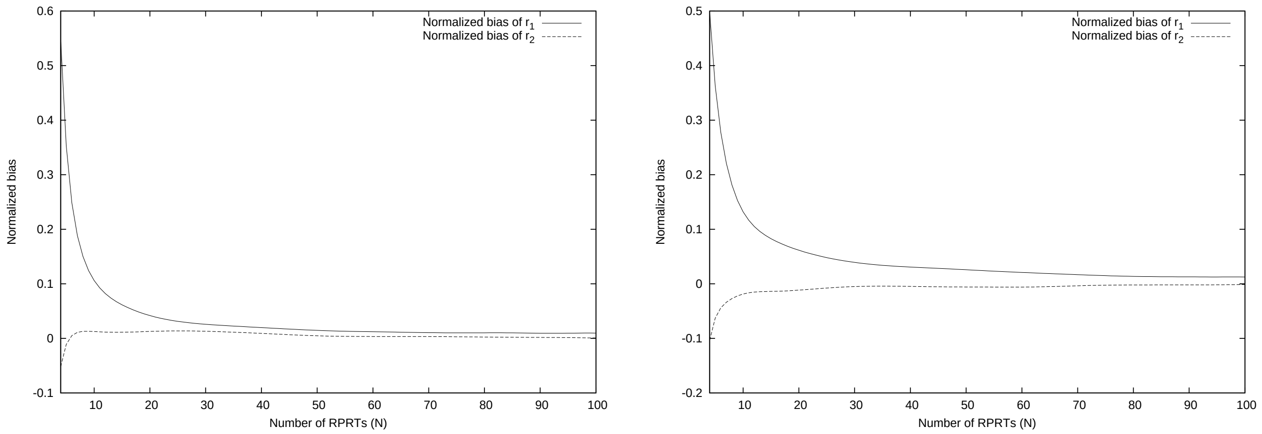
$$\begin{aligned}
E(\widehat{r_1}) &= \int_0^1 g(x)f_{X_N|r}(x)\,dx\\
&\overset{Eq.9}{\leq} -rN\binom{r-1}{N-1}\int_0^1\frac{x^{N-1}(1-x)^{r-N}}{\ln(1-x)}\,dx\\
&\quad +(N-1)\int_0^1 f_{X_N|r}(x)\,dx\\
&= -rN\binom{r-1}{N-1}\int_0^1\frac{x^{r-N}(1-x)^{N-1}}{\ln x}\,dx\\
&\quad +(N-1)\\
&\overset{Eq.12}{=} -rN\binom{r-1}{N-1}\sum_{i=0}^{N-1}(-1)^i\binom{N-1}{i}\\
&\quad \cdot\ln(r-(N-1)+i)+(N-1)\\
&\overset{Eq.15}{=} \frac{rN}{(N-1)!}\sum_{i=0}^{N-2}(-1)^i\binom{N-2}{i}\\
&\quad \cdot\left[\prod_{\substack{j=0\\ j\neq i}}^{N-2}(r-(N-1)+j)\right]\\
&\quad \cdot\ln\left(1+\frac{1}{r-(N-1)+i}\right)^{r-(N-1)+i}\\
&\quad +(N-1).
\end{aligned}$$

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,

$$\begin{aligned}
E(\widehat{r_1}) &\leq \frac{q\cdot r\cdot N}{(N-1)!}\sum_{i=0}^{N-2}(-1)^i\binom{N-2}{i}\\
&\quad \cdot\left[\prod_{\substack{j=0\\ j\neq i}}^{N-2}(r-(N-1)+j)\right]+(N-1)\\
&\overset{Eq.17}{=} q\cdot r\frac{N}{N-1}+(N-1)\\
&= q\cdot r\left(1+\frac{1}{N-1}\right)+(N-1). \qquad (18)
\end{aligned}$$

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

$$\begin{aligned}
p\cdot r\left(1+\frac{1}{N-1}\right) &\leq E(\widehat{r_1})\leq & q\cdot r\left(1+\frac{1}{N-1}\right)\\
&& +(N-1)\\
p\left(1+\frac{1}{N-1}\right) &\leq \tfrac{E(\widehat{r_1})}{r}\leq & q\left(1+\frac{1}{N-1}\right)\\
&& +\frac{N-1}{r}\\
\frac{p}{N-1}-(1-p) &\leq \tfrac{E(\widehat{r_1})-r}{r}\leq & \frac{q}{N-1}-(1-q)\\
&& +\frac{N-1}{r}. \qquad (19)
\end{aligned}$$



(a) $r = 1,000$ nodes

(b) $r = 10,000$ nodes

Fig. 1. The bias of $\widehat{r_1}$ and $\widehat{r_2}$ vs. N

Eq. 19 shows the upper and lower bounds on the bias 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 bias is approximately $\frac{1}{N-1}$.

Proof: The proof follows from Eq. 19. When $1 \ll N \ll r$, p and q are close to 1, and we have $\frac{E(\widehat{r_1})-r}{r} \approx \frac{1}{N-1}$.

■

Using the above analysis, we now present an enhanced gateway algorithm that removes the bias.

Algorithm 2: An enhanced gateway algorithm that removes the bias.

- Let $\widehat{r_1}$ be the estimation of Algorithm 1.
- An unbiased estimation is $\widehat{r_2} = \widehat{r_1} \cdot \frac{N-1}{N}$. ■

To see that $\widehat{r_2}$ is an unbiased estimator of r, note that $\frac{E(\widehat{r_2})-r}{r} = \frac{E(\widehat{r_1})\cdot\frac{N-1}{N}-r}{r} \approx \frac{\frac{N-1}{N}\cdot r(1+\frac{1}{N-1})-r}{r} = \frac{N-1}{N} + \frac{1}{N} - 1 = 0$.

Figure 1 shows Monte Carlo simulation results for the bias of $\widehat{r_1}$ and $\widehat{r_2}$ as found by Algorithm 1 and Algorithm 2 respectively. The results are given as a function of N for 1,000 and 10,000 affected nodes. For each value of N, each affected node drew a random number using a uniform distribution, and the smallest N numbers were used by the gateway as input to Algorithm 1 and Algorithm 2. Each such trial, with the same value of N, was repeated 1,000 times and the bias was computed. The graphs clearly show that $\widehat{r_1}$ has a positive bias whose value is significant for small values of N. In contrast, $\widehat{r_2}$ has no bias, and is therefore very accurate, even when N is very small.

We conducted additional Monte Carlo simulations to prove that Algorithm 2 is not only accurate, but also precise. To this end, we implemented two distribution functions: the uniform distribution and the truncated exponential distribution $f(x) = \frac{1}{e^{\lambda}-1} \cdot \lambda e^{\lambda x}$, for $x \in [0, 1]$. In the simulation, the affected nodes drew their random numbers from the above distributions and the N smallest numbers were used by the gateway as input to Algorithm 2. For each value of N, Figure 2 shows the percentage of trials, out of the 1,000 runs, for which $\frac{\widehat{r_2}}{r} \in (1-\varepsilon, 1+\varepsilon)$, where $\varepsilon = 0.05$, i.e., a confidence interval of 95%. The results are shown for a group of 1,000 affected nodes (Figure 2(a)) and for a group of 10,000 affected nodes (Figure 2(b)). Similarly, Figure 3 shows the percentage of trials, out of the 1,000 runs, for which $\frac{\widehat{r_2}}{r} \in (1-\varepsilon, 1+\varepsilon)$, where $\varepsilon = 0.03$, i.e., a confidence interval of 97%.

We conclude from these graphs that 30-40 response messages are sufficient to guarantee a confidence level of 95% with probability very close to 1. Similarly, to guarantee a confidence interval of 98% with probability very closed to 1, about 70 response messages are sufficient. Another conclusion is that it is better to use the truncated exponential distribution rather than the uniform distribution, especially when N is larger than 20. For smaller values of N, the uniform distribution gives better results.

V. Combining NATO! with a Collision Resolution Scheme

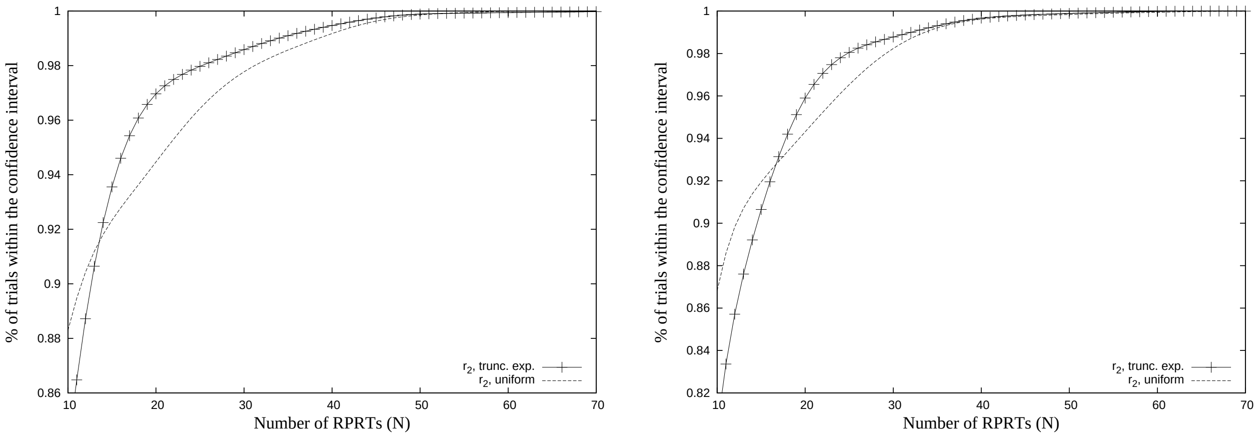
Losses of RPRT messages have a significant impact on the proposed estimation algorithm. In this section we first study this impact and then show how to overcome potential losses, either due to collisions in a shared channel or to transmission errors. Suppose that S of the first N RPRT messages are lost. Therefore, the RPRT considered by the gateway to be the N'th is actually the $N+S$'th. This influences Eq. 3, which the gateway solves to find r. This equation now should read:

$$\frac{1}{r} + \frac{1}{r-1} + \ldots + \frac{1}{r-(N-1)} = -\ln(1 - x_{N+S}). \quad (20)$$

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 message (as opposed to places where it indicates the number of such messages), we get:

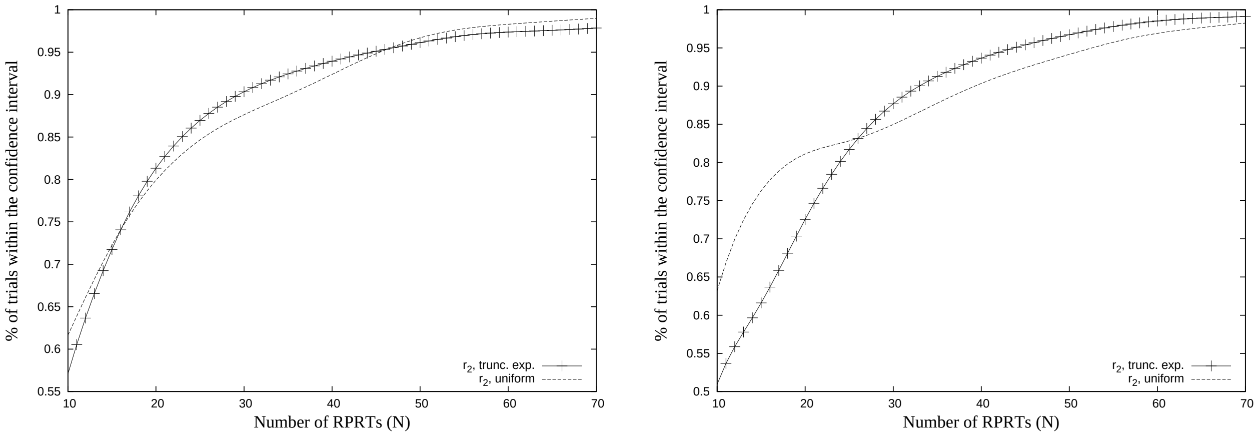
$$-p\frac{S-1}{N+S-1} - (1-p) \le \frac{E(\widehat{r_1})-r}{r} \le$$
$$-q\frac{S-1}{N+S-1} - (1-q) + \frac{N-1}{r}. \quad (21)$$

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



(a) $r = 1,000$ nodes (b) $r = 10,000$ nodes

Fig. 2. A Confidence Interval of 95%



(a) $r = 1,000$ nodes (b) $r = 10,000$ nodes

Fig. 3. A Confidence Interval of 97%

the bias of Algorithm 1 is $\frac{E(\hat{r_1})-r}{r} \approx -\frac{S-1}{N+S-1}$, and the bias of Algorithm 2 is $\frac{E(\hat{r_1})-r}{r} \approx -\frac{S}{N+S-1}$.

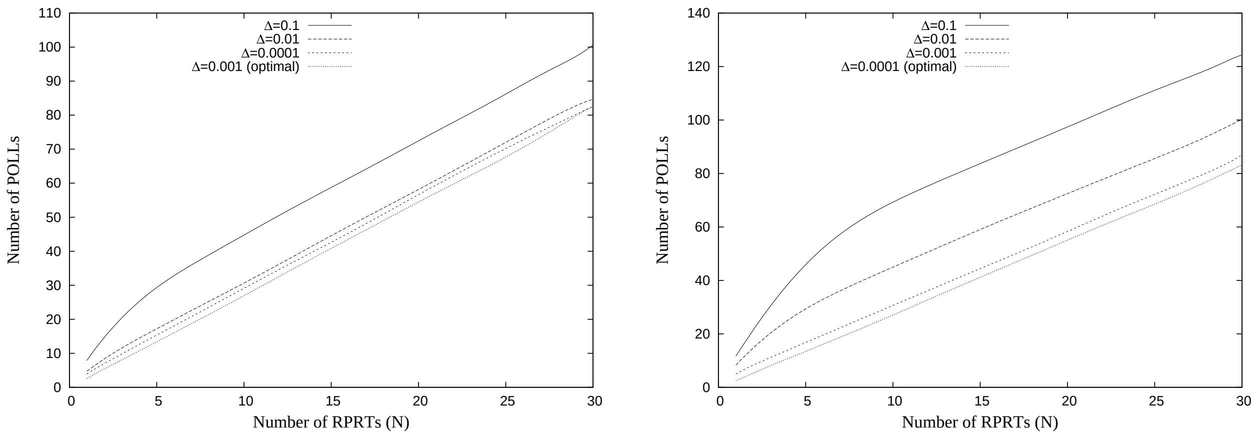
The above analysis clearly shows that the bias is as big as the loss rate of the RPRT messages. This would make NATO! impractical in environments where losses are possible either due to transmission errors or to collisions in shared channels. We now show how to efficiently implement NATO! in such environments. We consider a wireless network, such as 802.11 (in centralized mode), 802.16 or LTE, where the functionality of the gateway is fulfilled by the base station. The base station wants to use NATO! in order to estimate the number of nodes that experience some event. These nodes use the shared medium in order to send the RPRT messages to the base station.

The main idea is to run NATO! in conjunction with a distributed protocol for collision resolution [8], [17]. When a receiver needs to send a RPRT message, it draws a random number from the interval $[0, 1]$ using a uniform distribution. The base station needs to get only the RPRTs with the N smallest numbers. This is done by means of POLL messages broadcast by the base station on the downlink. Each such a message specifies an interval $(t_1, t_2]$. Receivers whose drawn number falls into this interval send their RPRTs in an uplink time slot specifically allocated by the base station for this purpose. Let the i'th POLL message be POLL(i). Let the interval specified by this message be $(t_1^i, t_2^i]$, where $t_2^i - t_1^i = \Delta^i$. The initial values are $t_1^1 = 0$ and $t_2^1 = \Delta^1 = \Delta$, where Δ is a constant of the algorithm. Hence, the interval announced by POLL(1) is $(0, \Delta]$. After the base station sends POLL(i), there are three possible cases:

[C1] No RPRT is received for $(t_1^i, t_2^i]$. In this case the base station sends the next POLL message, POLL$(i+1)$, with $t_1^{i+1} = t_2^i$ and $\Delta^{i+1} = 2\Delta^i$. In other words, the interval is shifted by Δ^i units and its width is doubled.

[C2] Exactly one RPRT is received, whose value is t. If this is the N'th RPRT to be received, the protocol stops. If this is not the N'th RPRT, the base station sends a new POLL message, POLL$(i+1)$, in order to obtain the next RPRT. For this POLL, $t_1^{i+1} = t$ and $\Delta^{i+1} = \Delta^i$. In other words the interval for POLL$(i+1)$ starts at t and has the same length as POLL(i). This is because this interval length is likely to contain exactly one RPRT in the next POLL too.

[C3] A collision occurs due to the transmission of two or more RPRTs. In such a case the base station makes



(a) 1,000 nodes

(b) 10,000 nodes

Fig. 4. Number of POLLs required for getting the smallest N RPRTs

a binary search for the first colliding RPRT on the interval $(t_1^i, t_2^i]$. If this RPRT is the N'th one, the algorithm stops. Otherwise, after this RPRT is found, say with the value t and by POLL(j), the algorithm sets $t_1^{j+1} = t$ and $\Delta^{i+1} = \frac{1}{2}\Delta^i$. In other words the interval for POLL$(i+1)$ starts at t and has half of the length of POLL(i). This is because this interval length is likely to contain less RPRTs than POLL(i), and with luck, exactly one RPRT.

The binary search on the interval $(a, b]$ works as follows. The sender sends a POLL, for which $t_1 = a$ and $t_2 = mid$, where $mid = \frac{1}{2}(a+b)$. Now there are several possible cases:

(a) If no RPRT is received, the binary search is recursively executed on the interval $(mid, b]$.

(b) If exactly one RPRT is received for the value t, the binary search stops and the algorithm searches for the next RPRT in $(t, t + \frac{1}{2}\Delta^i]$ as indicated above.

(c) If a collision of two or more RPRTs occurs, the binary search is executed recursively on the interval $(a, mid]$. ■

Every RPRT message successfully received by the base station has to be acknowledged. The acknowledgments can be piggybacked by the base station in the POLL messages. Still, it is possible that the base station might miss a relevant RPRT. For example, suppose that during the interval $(0, 0.5]$ two RPRT messages are sent: the first with 0.2 is sent by node v_1, and the second with 0.3 is sent by node v_2. Suppose also that the base station gets the RPRT(0.3) and not the RPRT(0.2) because v_2 is closer to the base station than v_1. Then, the base station will announce a new interval, such as $(0.5, 1]$, during which v_1 will not be allowed to send its RPRT message. To overcome this problem, we allow node v_1 to send its RPRT message during the latter interval as well. This will force the base station to get the RPRT of v_1 and to re-order the received messages.

The fact that the gateway uses multiple polling rounds to obtain the N smallest numbers does not imply that NATO! is a multi-round algorithm. All the nodes draw their random numbers only once, and the rounds are only needed in order to ensure reliable delivery of these numbers when collisions and errors are possible.

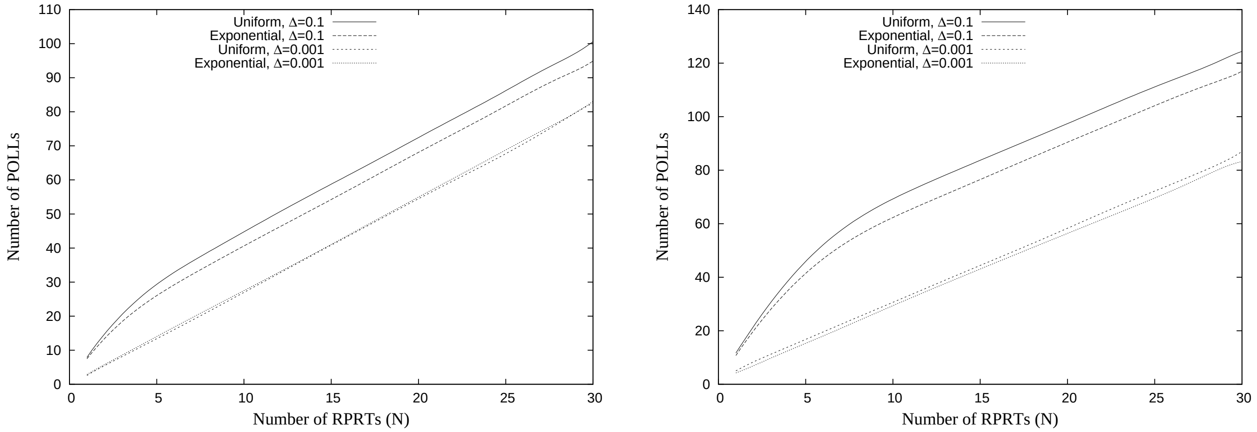
We now present simulation results for the performance of the above protocol. The bandwidth and the time consumed by this protocol are both functions of the number of POLL messages, because each POLL requires one round trip and one uplink slot. Hence, we measure the performance of this scheme in terms of this number.

We simulate the scheme by having r receivers, from which the base station needs to receive N RPRTs. To receive these RPRTs, the base station sends POLL messages as dictated by the scheme. This procedure is repeated 50 times for every value of N. Our graphs depict the average number of POLLs during these 50 runs as a function of N. In all the graphs the x-axis is N and the y-axis is the number of POLLs.

Figure 4 depict several curves with different initial values of Δ when the reporting nodes choose their times using a uniform distribution. These values are relevant only when the algorithm starts running. We can see that when N becomes larger, the interval size is likely to adapt to the number of contending users. Hence, all the curves have the same slope.

While $\Delta = 1/r$ will result in fewer POLLs than other values of Δ, we cannot select such a value since r is unknown. From Figure 4(a) we see that it is better to choose a value smaller than $1/r$ rather than a value bigger than $1/r$. Since we are interested only in the first N sequence numbers, a good distribution should ensure that RPRTs are less frequent at the beginning of the interval, perhaps at the expense of being more frequent near its end. An example of such a distribution is the truncated exponential distribution on the interval $[0, 1)$: $f(x) = \frac{1}{e^\lambda - 1} \cdot \lambda e^{\lambda x}$, where $\lambda > 0$.

Figure 5 shows the number of POLL messages needed for N RPRTs with truncated exponential distribution and $\lambda = 1$, compared to the uniform distribution, for $r = 1,000$ and $r = 10,000$ nodes. Four curves are shown in each figure. Consider first the bottom two curves, which represent the case where $\Delta = 1/r$ (in Figure 5(a) they almost fully overlap). The upper one is for the uniform distribution while the lower is for the exponential distribution. For this setting, the exponential distribution performs similarly or only marginally better than the uniform one, because $\Delta = 1/r$ is a good choice, as



(a) 1,000 receivers

(b) 10,000 receivers

Fig. 5. Number of POLLs required for getting the smallest N RPRTs using a uniform and a truncated exponential distribution

explained above. On the other hand, selecting a bad value for Δ, such as $\Delta = 0.1$ (the top two curves in Figure 5(a) and 5(b)), shows a clear improvement of the exponential distribution over the uniform one.

VI. Conclusions

We presented a new probabilistic polling scheme for estimating the size of a group of nodes affected by the same event. The proposed scheme is generic in the sense that it does not depend on the physical characteristics of the underlying network. The algorithm is based on the N minimum sequence numbers drawn by the nodes. It defines the likelihood function for the received RPRTs and then uses the Newton-Raphson method to find the number of receivers for which this function is maximized. We analyzed the bias of our algorithm and showed that it approximately equals $1/(N-1)$. We used this important result to correct the algorithm and bring its bias to 0. We showed that the algorithm performs very well in terms of the number of response messages needed in order to guarantee a confidence level of 95% or 97% with probability very close to 1. Finally, we showed how the proposed algorithm can be combined into a collision resolution scheme in order to guarantee its reliability in unreliable networks.

References

[1] B. Adamson, C. Bormann, M. Handley, and J. Macker. Negative-acknowledgment (NACK) - oriented reliable multicast (NORM) protocol. RFC-3940, November 2004.

[2] S. Alouf. Parameter estimation and performance analysis of several network applications, November 2002.

[3] S. Alouf, E. Altman, C. Barakat, and P. Nain. Estimating membership in a multicast session. In *SIGMETRICS*, 2003.

[4] S. Alouf, E. Altman, and P. Nain. Optimal on-line estimation of the size of a dynamic multicast group. In *INFOCOM*, 2002.

[5] M. Ammar and G. Rouskas. On the performance of protocols for collecting responses over a multiple-access channel. 1991.

[6] J. Bolot, T. Turletti, and I. Wakeman. Scalable feedback control for multicast video distribution in the internet. In *SIGCOMM*, 1994.

[7] C. Budianu, S. Ben-David, and L. Tong. Estimation of the number of operating sensors in large-scale sensor networks with mobile access. *IEEE Transactions on Signal Processing*, 54(5), 2006.

[8] J. I. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, 25:505–515, September 1979.

[9] F. de Belleville, L. Dairaine, C. Fraboul, and J.Y. Tourneret. Group size estimation for hybrid satellite/terrestrial reliable multicast. In *Broadband Satellite Comunication Systems and the Challenges of Mobility, IFIP International Federation for Information Processing*, 2005.

[10] T. Friedman and D. Towsley. Multicast session membership size estimation. In *INFOCOM*, 1999.

[11] C. Liu and J. Nonnenmacher. Broadcast audience estimation. In *INFOCOM (2)*, 2000.

[12] J. Nonnenmacher and E. Biersack. Optimal multicast feedback. In *INFOCOM*, 1998.

[13] J. Nonnenmacher and E. Biersack. Scalable feedback for large groups. *IEEE/ACM Transactions on Networking*, 7(3), June 1999.

[14] J. Nonnenmacher, E. Biersack, and D. Towsley. Parity-based loss recovery for reliable multicast transmission. *IEEE/ACM Transactions on Networking*, 6(4):349–361, 1998.

[15] S. C. Port. Theoretical probability for applications, Wiley-Iterscience, 1994.

[16] D. Rubenstein, J. Kurose, and D. Towsley. A study of proactive hybrid FEC/ARQ and scalable feedback techniques for reliable, real-time multicast. *Computer Communications*, 24(5–6):563–574, 2001.

[17] B. S. Tsybakov and V. A. Mikhailov. Slotted multiaccess packet broadcasting feedback channel. *Problemy Peredachi Informatsii*, 14:32–59, October - December 1978.

[18] A. Wood and J. Stankovic. Denial of service in sensor networks. *IEEE Computer*, 35(10):54–62, October 2002.