Distributed Adaptive Protocols for Information Dissemination in Large-Scale Communication Systems

Sachin Shetty
Rowan University
Glassboro, NJ 08028, USA
shetty@rowan.edu

Min Song, Jun Wang
Old Dominion University
Norfolk, VA 23529, USA
{msong, jwang012}@odu.edu

***Abstract*—One approach for information dissemination in large-scale communication systems is using epidemic protocols. Current epidemic protocols, however, adopt a constant fanout policy, which does not enable end users to control the information dissemination process. For distributed applications that need to compute a global function within a pre-determined response time, better procedures to control the information dissemination process have to be developed. In this paper, we introduce two distributed adaptive epidemic protocols using a dynamic fanout scheme. They are named *Round-Based dynamic fanout* (RBdf) and *Cluster-Based dynamic fanout* (CBdf). In RBdf, the network topology is flat and each node transmits a message with a varied fanout every round. In CBdf, the network topology is hierarchical, and the fanout values in every cluster differ within the same round. The main objectives are to ensure that peers receive messages within a bounded latency and that the system message overhead is a bounded value. The performance of the proposed protocols are verified through both theoretical and simulation studies.**

1. INTRODUCTION

One approach for information dissemination in large-scale communication systems is using epidemic protocols. An epidemic protocol proceeds through asynchronous rounds before the information is reliably disseminated to every node. A round is defined as the time taken for all nodes to disseminate a message to their neighboring nodes. In the basic epidemic protocol, every node within the system is potentially involved in the information dissemination process. A peer node that has delivered a given message will be termed *infective*, otherwise *susceptible*. Basically, every node buffers every message received up to a certain buffer capacity and forwards every message a limited number of rounds. The node forwards the message each time to a randomly selected set of nodes. The size of this set is called *fanout*. The use of epidemic protocols has been explored in applications such as reliable multicast [14], failure detection [17], data aggregation [7,10], resource discovery and monitoring [18], and database replication [4]. Each of these applications implements different variations of the basic epidemic protocol. For instance, in [10], the aggregation protocol is based on the simple "push-pull gossiping" scheme. In this scheme, every node executes two different threads. The active thread periodically initiates an *information exchange* with one random neighbor. The passive thread waits for messages sent by one sender. We observe that in most of the epidemic protocols, fanout is kept a constant which is based on the assumption that every peer possesses uniform and independent membership. This assumption is not valid in practical applications where every peer cannot be expected to be infected independently of other peers. Moreover, a constant fanout is only applicable in scenarios where the total number of infected nodes in every round is immaterial. But this may not be the case wherein a controlled infection pattern is expected. The infection pattern is a frequency distribution specifying the number of nodes that are expected to be infected at the end of a round. By controlling the infection pattern, the end user has a better control over the overall latency and message overhead of the information dissemination process.

In this paper, we introduce two distributed adaptive epidemic protocols using a dynamic fanout scheme. They are named *Round-Based dynamic fanout* (RBdf) and *Cluster-Based dynamic fanout* (CBdf). In RBdf, the network topology is flat and each node transmits a message with a varied fanout every round. The fanout values are quantified based on the infection pattern and redundancy message pattern over rounds, but the fanout remains constant within a round. In CBdf, the network topology is hierarchical, and the fanout values in every cluster differ within the same round. Nodes are clustered based on a geographic proximity criterion and fanouts vary between clusters of nodes. This implies that during each round, nodes in different clusters disseminate information using different fanout values. In both approaches, the number of messages generated is bounded by $O(n \log n)$, where n is the total number of nodes in the system. In spite of ensuring user-controlled information dissemination, the lower bound on the message overhead for both the approaches is the same as the one achieved in epidemic protocols with a constant fanout [3,14].

The rest of the paper is organized as follow. In Section 2, we present the related work for epidemic protocols. Section 3 introduces RBdf, and Section 4 presents CBdf. For both protocols, theoretical analysis is provided. Section 5 gives simulation results supporting the analysis. Section 6 concludes the paper and discusses the future work.

2. RELATED WORK

As mentioned in the introduction section, most of the epidemic protocols in the literature usually adopt a constant fanout [2, 17, 5, 13]. The Bimodal multicast scheme [2] and probabilistic multicast [17] schemes adopt a constant fanout of 1. The gossip protocols for publish/subscribe systems [5] specify constant fanout values based on network size. In particular, the fanout values are increased as the network size increases. For systems with dynamic behavior where information is changing continuously, a spatial epidemic protocol [13] bounds propagation time by a poly-logarithmic function in distance by choosing epidemic targets with a probability which is an inverse polynomial function of distance. The fanout is assumed to be a constant of one. In [12], it is discussed that a generic gossip protocol needs $O(n \log n)$ messages to spread a rumor.

The very first attempt to compute an optimal value for the fanout in a probabilistic reliable information dissemination process was performed by [14]. They computed the fanout needed to deliver information to all nodes with a high probability by using random graphs. They show that a fanout in the order of $\log(n) + c + o(1)$ gives a success probability of $e^{-e^{-c}}$, where c is a design parameter whose value ranges from 0 to 1. The reliability of this gossip-based protocol is related to key system parameters (system size, failure rates, and the number of gossip targets).

3. ROUND BASED DYNAMIC FANOUT

The basic idea underlying our approach, inspired by the work presented in [5], is as follows. Each peer maintains a fixed size view of member peers. This view is sorted according to their network distance estimates. Therefore, the first position in the view holds the closest peer known so far. During the protocol initialization phase, views need to be initialized with a random sample of nodes taken from the whole peer-to-peer network. For this purpose, we use Newscast [11] to build and maintain an approximately random-graph overlay topology. In order to evolve the topology, peers exchange views in an epidemic fashion. Periodically, each peer actively selects a neighbor from its view and starts a view exchange process (see pseudo-code in Fig. 1). Each peer node execution implements two threads. The active thread is shown in Fig. 1a, and the passive thread is shown in Fig. 1b. In the active thread, every peer node picks the set of random neighbors based on the round-based fanout. The round-based fanout is computed based on user specifications, which are specified as an infection pattern over rounds. The peer node sends the message and its local view to this set of random neighbors. Once the remote peer's view has been received, it is merged with the local one. This merge operation preserves the ordering of the local view, i.e., newly received member peers are sorted in accordance with their network distance estimates.

```
do forever
    // Wait for finite interval of Δt, which is equal to the time taken
    for a round to be completed
    wait(Δt);
    // Roundfanout computes the fanout based on the user specified
    infection pattern to be used in the current round.
    fanout= Roundfanout(current round);
    // The neighbor list is populated using the SELECTPEERS
    method.
    Neighbors[fanout] = SELECTPEERS();
    // Message is transmitted to the neighbor list using the
    SENDMESSAGE method.
    SENDMESSAGE(Neighbors[fanout]);
    // The view of the current node is sent to the neighbors list
    SENDSTATE(myview, Neighbors[fanout]);
    // Receive the view from the node who sent its view
    n_state = RECEIVESTATE();
    //Merge the current view with the received view in a temporary
    list.
    my_state.UPDATE(n_state);
```

(a) Active Thread

```
do forever
    n_state = RECEIVESTATE();
    SENDSTATE(n_state.sender);
    my_state.UPDATE(n_state);
```

(b) Passive Thread

Figure 1: Round based Dynamic Fanout pseudo-code.

Notations

n	Number of peers in a peer-to-peer network
r	A single epidemic round
$R_{\max}$	The maximum number of rounds
$\in$	Probability of a message loss
k	Number of peer crashes in a round
τ	Probability of a peer crash during a single round. $\tau = k / n$
β_r	Probability of redundant messages during round r
I_r	Number of susceptible peers that are infected by a message sent from an infective during round r. $I_0 = 1$, indicating at round 0 there is only one peer with a message.
S_r	Number of peers that are not infected in the network after the end of round r. $S_0 = n - 1$ and we expect the epidemic period to end with a high probability of $S_{R_{\max}} = 0$.
F_r	The fanout associated with every peer, i.e., a peer that receives a message during round r will transmit the message to F_{r+1} peers in round $r+1$.

In the following analysis, we assume that the composition of the network does not vary during the run, and we observe the

transmission of a single message from a peer. Each peer participates in the gossip process via synchronous rounds. During each round, each peer has an independent, uniformly distributed random view of known peers. Thus, peers have a global membership view, and epidemic targets are picked from this global view uniformly and randomly. We also assume two kinds of failures affecting our system. They are *message loss* and *peer crash*. Both failures are assumed to be stochastically independent. All nodes are assumed to have the same failure probabilities. The values of τ and $\in$ are the same as the corresponding values in [5]. The recovery of crashed peers is not taken into consideration, nor Byzantine failures. We assume that redundant messages are generated every round. We estimate the redundant message factor β_r from our simulations discussed in the next section.

Mathematical Analysis

The main objective of our analysis is to measure the adaptive fanout value in every round and the number of messages generated after the epidemic protocol successfully terminates. The first step of our analysis is to estimate the distribution of S_r and β_r over the $R_{\max}$ rounds. We plan to associate values for both distributions with an exponential rule. The constraints of the exponential rule are: $S_0 = n-1, S_{R_{\max}} = 0$, and $\beta_0 = 0, \beta_{R_{\max}} = thresh$ where *thresh* is the maximum probability of redundancy allowed in the network. After estimating S_r and β_r, the corresponding fanout values for each round *r* can be determined.

Our analysis is based on the chain-binomial based recurrence relation [1,3], which has been derived from epidemic models and successfully applied to epidemic protocols in peer-to-peer networks. From [5], the lower bound on the probability that a given susceptible peer is infected by a message is given by

$$p = \left(\frac{F}{n-1}\right)(1-\in)(1-\tau)$$

where F is the constant fanout size. In our approach, the fanout varies from round to round. We conjecture that the equation for probability of infection p_r during individual rounds remains the same. In the analysis provided in [5] redundant or duplicate messages were assumed to be discarded by peers, and their impact on the overall message overhead was not considered. After incorporating the variable fanout F_r and the probability of redundancy in the above equation, the probability that a given susceptible peer is infected by a given message in round *r* is given by:

$$p_r = \left(\frac{F_r}{n-1}\right)(1-\in)(1-\tau)(1-\beta_r) \quad (1)$$

Let $q_r = 1 - p_r$ be the probability that a given susceptible peer is not infected by a given gossip message in round *r*. Also the probability that a given susceptible peer is not affected by the presence of I_r infected peers is $q_r^{I_r}$. On the same lines, we can derive that the number of peers which would not be infected by the message in the round *r*+1 is:

$$S_{r+1} = S_r q_r^{I_r} \quad (2)$$

Substituting (1) into (2) we get

$$S_{r+1} = S_r\left[1-\left(\frac{F_r}{n-1}(1-\in)(1-\tau)(1-\beta_r)\right)\right]^{I_r} \quad (3)$$

After some algebraic manipulations, the fanout value for round *r* can be computed by

$$F_r = \left[\frac{n-1}{(1-\in)(1-\tau)(1-\beta_r)}\right]\times\left[1-\exp\left(\frac{\ln\left(\frac{S_{r+1}}{S_r}\right)}{I_r}\right)\right] \quad (4)$$

To compute the value of I_r, we use the following simple relationship

$$I_r = S_{r-1} - S_r \text{ , for } 1 \le r \le R_{\max} \quad (5)$$

The fanout values computed for every round will ensure that that the message will reach every peer. Next we proceed to compute the lower bound on the message overhead of our adaptive fanout approach. The delay involved in the fanout calculation is only in the initial setup of the values for S_r by the user. Once the user specifies the expected values for S_r, there is no additional involvement of the user in the fanout calculation.

Message Overhead

In our approach, the number of messages generated during each round by a peer is not constant and is determined by the value of F_r. At the beginning of round *r*, each peer transmits F_r new messages into the network, and I_r peers are participating in the message transmission process. So the number of new messages generated in round *r* is $F_r \times I_r$. We have $n >> F_r$ in practical peer-to-peer systems. Thus (3) becomes:

$$S_{r+1} = S_r \times \exp\left[-\left(\frac{F_r}{n-1}(1-\in)(1-\tau)(1-\beta_r)\right)\times I_r\right] \quad (6)$$

Applying logarithms and algebraic manipulations, we get an expression for $F_r \times I_r$ as follows,

$$F_r \times I_r = \left[\frac{n-1}{(1-\in)(1-\tau)(1-\beta_r)}\right]\times\log\left(\frac{S_r}{S_{r+1}}\right) \quad (7)$$

Summing up (7) for all permissible rounds, the total number of messages generated in the network is given as:

$$M_{R_{\max}} = \left[\frac{n-1}{(1-\in)(1-\tau)(1-\beta_r)}\right]\times\sum_{r=0}^{R_{\max}-1}\log\left(\frac{S_r}{S_{r+1}}\right) \quad (8)$$

After the summation process, (8) reduces to

$$M_{R_{\max}} = \left[\frac{n-1}{(1-\in)(1-\tau)(1-\beta_r)}\right]\times\log\left(\frac{S_0}{S_{R_{\max}}}\right) \quad (9)$$

Eq. (9) can be approximated as

$$M_{R_{\max}} = \left[\frac{n-1}{(1-\in)(1-\tau)(1-\beta_r)}\right]\times\log(n-1) \quad (10)$$

Thus, the message overhead is bound by $O(n \log n)$, which is similar to (the same as) other constant fanout based epidemic protocols [3, 14].

4. CLUSTER BASED DYNAMIC FANOUT

In this section, we present the Cluster-Based dynamic fanout based epidemic protocol, followed by a detailed mathematical analysis. We adopted a hierarchical-membership approach using Newscast [11]. The nodes do not perform message filtering which means that redundant and duplicate messages are processed during every epidemic instance. In the Round-Base dynamic fanout protocol, nodes were connected by a flat membership where the local subscription list is composed of nodes located all over the network. The fanout for a given node changes from round to round, but remains constant within a round. In our Cluster-Base dynamic fanout protocol, we adopt a cluster model where nodes are clustered according to a geographical proximity criterion. The pseudo-code for the information dissemination process in this approach is similar to Fig. 1. The only difference is the fanout values and the view list for each node.

In the Cluster-Based dynamic fanout approach, the probability p that a given susceptible peer is infected by a message follows a frequency distribution during every round. There are two ways in which p can be varied: vary p among each node in the network or vary p between clusters. We chose to vary p between the clusters. The reasoning for this choice is as follows. Our network topology consists of heterogeneous clusters, which may include workstation clusters composed of machines with different processor architectures, data formats, and operating system environments. It is arguable that the probability of infection should vary between the nodes within a cluster too, but such a level of detail may make the model intractable. In the following analysis, we present an agreement between our hypothesis and observed data.

The network topology is shown in Fig. 2. The figure shows a two level hierarchy for ease of analysis, and can be easily extended to a hierarchy of more levels. The figure shows two types of nodes: Empty circles represent internal nodes, while solid circles represent external nodes. Internal nodes have a local subscription list composed exclusively of nodes belonging to the same cluster; external nodes are provided with remote subscription list consisting of nodes in other clusters. The presence of these two types of nodes in our network topology leads to two kinds of fanout: intra-cluster fanout and inter-cluster fanout. The intra-cluster fanout denotes the constant number of links each internal node has with member nodes in the same cluster. In our topology, this fanout remains constant within each cluster. But the inter-cluster fanouts vary between any given pair of clusters. In Fig.2, clusters C_1 and C_2 have a fanout of 2, while clusters C_3, C_4, and C_5 have a fanout of 3. The inter-cluster fanout denotes the number of remote links each external node must maintain with other external nodes. In our topology, this fanout is a constant of 1.

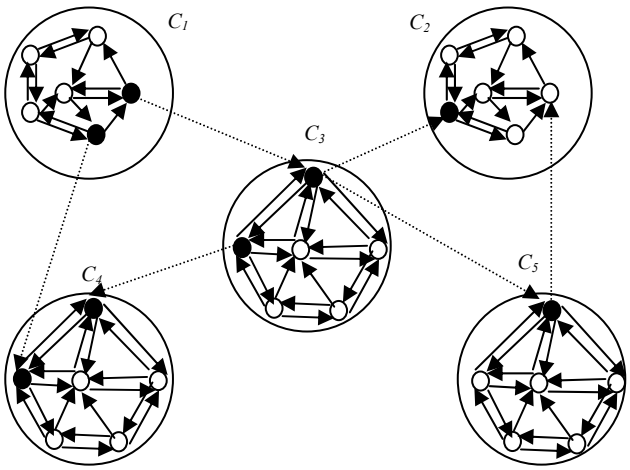

Figure 2: Cluster membership depicting intra-cluster fanout and inter-cluster fanout.

Analytical Model

The basis for our analytical model is the variation of the inter-cluster fanout. To model this variation we focus on varying the value of p. There are different ways to model the variation of p. We choose to adopt a simple way to take account of variation of p, by introducing a suitable distribution for p and then go through the process of estimating the parameters in the distribution and testing goodness-of-fit. We assume p is a random variable following a beta distribution. We prefer to use a beta distribution because an exhaustive analysis would lead to a very complex Markov Chain with an impractical size. Beta distribution is used to model events where the random variable varies between two extremes. In our analysis, p varies between zero and total neighbor size.

As our analyses are based on the *chain-binomial model* [5], we define the term "chain" before we proceed with our analysis. At each stage of an epidemic, there are certain numbers of infective and susceptible nodes, with the latter yielding new infective nodes at next stage distributed in a binomial series. Thus, we have a chain of binomial distributions. An instance of a chain in a cluster is denoted by:

$$(S,I)_{[0,R_{\max}]} \equiv \{(S,I)_r : r = 0,1,...,R_{\max}\}.$$

For example, if all the nodes in a 5-node cluster have been infected in 3 rounds then a possible chain sequence would be $(4,1)_{[0,3]} \equiv \{(4,1)_0,(2,3)_1,(1,4)_2,(0,5)_3\}$.

The probability of occurrence of a chain in a cluster can be expressed as follows:

$$P(p) = P((S,I)_{[0,R_{\max}]} \mid p). \quad (11)$$

Since we assume that p varies between clusters only, we compute the required expectations by averaging the

probability of frequencies of every kind of chain $P(p)$ over all possible values of p in the interval (0, 1), which gives the following,

$$\int_{p=0}^{1} P(p) f(p) dp. \tag{12}$$

The density function is given by:

$$f(p) = \frac{p^{a-1}(1-p)^{b-1}}{B(a,b)}, 0 \le p \le 1, \tag{13}$$

where $B(a,b)$ is the beta function, $a > 0,$ and $b > 0.$

The probability $P(p)$ is derived from the chain-binomial model as:

$$P(p) = P(S_{r+1} = j \mid S_r = i) = \binom{n-i}{j-i} (1-q^j)^{j-i} q^{j(n-j)}, j \ge i. \tag{14}$$

In (14), $q = 1 - p$ represents the probability that a susceptible peer is *not* infected by a given message. By integrating over all values of p where p has the beta density in (13), the expectations can be shown as functions of parameters *a, b, i ,j*. The probability $P(p)$ of this chain equals

$$\begin{aligned} P(p) &= \prod_{r=1}^{R_{\max}} \binom{n-s_r}{s_{r+1}-s_r} (1-q^{s_r})^{s_{r+1}-s_r} (q^{s_r})^{n-s_{r+1}} \\ &= \left[\prod_{r=1}^{R_{\max}} \binom{n-s_r}{s_{r+1}-s_r} (1-q^{s_r})^{s_{r+1}-s_r} \right] q^{\sum_{r=1}^{R_{\max}} s_r (n-s_{r+1})} \\ &= \left[\prod_{r=1}^{R_{\max}} \binom{n-s_r}{s_{r+1}-s_r} \sum_{m=0}^{n-s_{r+1}} \binom{n-s_{r+1}}{m} (-1)^m q^{m s_r} \right] q^{\sum_{r=1}^{R_{\max}} s_r (n-s_{r+1})}. \end{aligned} \tag{15}$$

Combining (13) and (15), and integrating $P(p)$ over all values of p, we have

$$\begin{aligned} E[P(p)] &= \int_0^1 \binom{n-s_r}{s_{r+1}-s_r} (1-q^{s_r})^{s_{r+1}-s_r} (q^{s_r})^{n-s_{r+1}} \frac{(1-p)^{a-1} p^{b-1}}{B(a,b)} dp \\ &= \binom{n-s_r}{s_{r+1}-s_r} \sum_{k=0}^{n-s_{r+1}} \binom{n-s_r}{k} (-1)^k \int_0^1 \frac{p^{s_r(k+s_r)+b-1}(1-p)^{a-1}}{B(a,b)} dp \\ &= \binom{n-s_r}{s_{r+1}-s_r} \sum_{k=0}^{n-s_{r+1}} \binom{n-s_r}{k} (-1)^k \frac{B(s_r[k+s_r]+b, a)}{B(a,b)} \end{aligned} \tag{16}$$

Equation (16) provides a mechanism to measure the expected value of the probability of a particular chain. The next step is to estimate the parameter values of *a,* and *b*. We estimate these values based on the data collected from simulation experiments. For example, in one sample simulation scenario, we setup a network topology of 100 clusters, each with a membership of 5 nodes. The initial sequence in each cluster would be comprised of the chain $S_0 = 4$ and $I_0 = 1$. In our example, the expected number of clusters with the chain (1, 3, 0) equals

$$\begin{aligned} &100 \int_0^1 4 p^4 (1-p)^3 \frac{(1-p)^{a-1} p^{b-1}}{B(a,b)} dp \\ &= 400 \left[\frac{B(a+3, b+4)}{B(a,b)} \right] \end{aligned} \tag{17}$$

Similar integrations for other, more probable chains based on the simulation data will be carried out, and the estimation of the parameters *a* and *b* will be done by applying maximum likelihood methods.

5. SIMULATIONS AND ANALYSIS

In this section, we compare the analytical results obtained for both approaches with simulation results. For the round based dynamic fanout approach, our results highlight the impact on infection pattern and message overhead in the presence of variable fanouts from round to round. For the cluster based dynamic fanout approach, our results show the relation between the infection pattern of clusters and the beta-distribution characterized probability of infection model. We conducted the simulations using peersim, an open source peer-to-peer simulator developed at the University of Bologna [9,16].

5.1 Round Based Dynamic Fanout

In our simulations, we have used Newscast as the underlying overlay network membership protocol [11]. The reason for this choice is twofold: First, we want to show empirical results in a realistic overlay network that can actually be built in a decentralized way. Second, Newscast is known to be robust and capable of maintaining a sufficiently random network in failure scenarios. In Fig. 1a, Newscast has been used to implement the SELECTPEERS function. We have performed our simulations on network sizes ranging from 500 nodes to 2,500 nodes. The size of the local neighbor sets at each peer node, which are maintained and exchanged by the NEWSCAST protocol, is set to 1% of the overall network size. This value is large enough such that a given message is able to reach all nodes in the network. For our approach to be effective, we rely on the user for the infection pattern. In our simulations, we have tackled the issue of designing good user inputs by adopting an exponential distribution rule. Using this rule, the distribution of the non-infection pattern S_r and the redundancy probability β_r was computed. In computing the distribution for β_r, the *thresh* value is set to 0.3. The values for τ and $\in$ are set to 0.05 and 0.01, respectively. We have conducted 20 simulation runs. Using equation (4), the fanout values were computed for each round. To incorporate fractional values in our simulation runs, we converted the real number to a lower or higher integer value randomly in different runs such that the average value is equal to the real number. Table 1 presents the analytical and simulation values for the non-infection pattern S_r with network sizes of 1,000 and 2,000 nodes. As is evident from Table 1, the option of variable fanouts from round to round allows users to control the infection pattern. In addition, the user can specify the total number of rounds for a specific infection pattern, and the fanouts are computed accordingly. Though the S_r values for individual runs deviated from their corresponding analytical values, the average over 20 runs converged to the theoretical values. Table 1 summarizes the average values for S_r over a set of runs and the standard deviation of the observed simulation

results. The simulation values for S_r do confirm to our analytical model.

Table 2 presents the computed dynamic fanout values for network sizes of 1,000 and 2,000 nodes. The fanouts are computed for each round. Table 3 highlights the message overhead in the round-based dynamic fanout approach under different network sizes. The third column shows the cumulative message overhead at the end of all rounds generated by the simulations. Column 4 shows the theoretical values for the message overhead computed based on the analytical results from Eq. (10). We observed that the overall message overhead is independent of the variations in fanout but is dependent on the percentage of redundant messages in the network. We observe from Table 3 that as the number of nodes increases, the message overhead deviates from the analytical values. It is interesting to notice that the ratio of simulation message overhead to analysis one is almost constant (1.18) for different number of nodes. This deviation is largely due to the fact that as network size gets larger, the percentage of redundancy messages also increases.

Table 1. Round based dynamic fanout for 1,000 and 2,000 nodes

Round	1000			2000		
	Sr Analysis	*Sr* (avg) Simulation	Std Dev	*Sr* Analysis	*Sr* (avg.) Simulation	Std Dev
1	999	999	0	1999	1999	0
2	994	992.4	0.7	1994	1991.5	0.7
3	961	959.3	0.5	1961	1959.4	0.5
4	771	770.5	0.7	1728	1725.3	0.7
5	210	209.5	2.6	969	965.5	2.6
6	1	0.5	0	52	50.3	0
7				1	0.5	

Table 2. Round based dynamic fanout for different network sizes

Rounds	Nodes				
	500	1000	1500	2000	2500
1	-	-	-	-	-
2	5	5	5	5	3
3	4.089	4.46	4.44	4.27	3.5
4	4.47	5.79	13.68	10.848	3.785
5	7.76	6.32	2.33	3.08	7.33
6	10.86	8.37	4.94	5.56	4.19
7	-	-	-	-	3.4016
8	-	-	-	-	4.99
9	-	-	-	-	15.875

5.2 Cluster Based Dynamic Fanout

For this approach, we have run experiments on two scenarios. In the first scenario, our network topology comprised of 100 clusters, with each cluster consisting of four nodes. In the second scenario, the total number of clusters remained the same, but the number of nodes in each cluster was increased to five. As the infection probability varies for each cluster, we observe the infection pattern over rounds. The infection pattern for each cluster is represented as a binomial chain.

Table 3. Message Overhead in the Round based Dynamic Fanout approach

Number of Nodes	Maximum Rounds	Message Overhead (simulations)	Message Overhead (Analysis)
1,000	6	3540	3000
1,500	7	5624	4764
2,000	7	7795	6602
2,500	9	10031	8494

For example, in the first scenario, a cluster with four nodes where one node is infected ($I_0 = 1$) and three nodes are not yet infected ($S_0 = 3$), we recorded the number of new infected nodes $\{I_r\}$ at $r = 0, 1, 2, \ldots$ rounds as the chain 1-1-2, where all nodes are eventually infected. In Table 4, the first column shows all the possible infection pattern chains for a cluster of size four. The chain-binomial probabilities for all these chains $\{I_0 = 1, I_1, \ldots, I_k\}$ are given in the second column, where *n* is the total number of clusters. Our goal is to find out the total number of clusters which have the same infection pattern. Each entry in column two shows the total number of clusters for the corresponding chain in the first column. We need estimates of *p* and *z* to find out the total number of clusters for each infection pattern chain. The third column shows the expected probabilities of the chain-binomial model. The expected probabilities are presented by simpler notations as follows:

$$\begin{aligned} z &= (a+b)^{-1} \\ p &= a/(a+b) \\ z_1(n) &= \prod_{i=0}^{n} (1+iz) = z(n) \\ z_q(n) &= \prod_{i=0}^{n} (q+iz) \\ z_p(n) &= \prod_{i=0}^{n} (p+iz) \end{aligned} \tag{19}$$

Table 4. Chain-Binomial Probabilities for cluster size of 4

Infection Pattern $\{I_r\}$ $r = 0\ 1\ 2\ 3$	Probabilities	Expected Values of Probabilities
1-1-1-1	$6np^3q^3$	$6z_q(2)z_p(2)/z(5)$
1-1-2	$3np^3q^2$	$3z_q(1)z_p(2)/z(4)$

1-2-1	$3np^3q$	$3z_q(0)z_p(2)/z(3)$
1-3	np^3	$z_p(2)/z(2)$

In Table 5, the first column presents the binomial chains, and the second column contains the total number of clusters from simulation results. The third column contains the total number of clusters which need to be computed based on the estimates of p and z. To find the estimates of p and z, we employ a log-likelihood function. The log-likelihood function is constructed based on the expected probabilities in the third column of Table 4 and the simulation-generated total number of clusters in the second column of Table 5. The log-likelihood function $Log\ L$ is given as follows,

$$LogL = 5*\log(6z_q(2)z_p(2)/z(5)) + 6*\log(3z_q(1)z_p(2)/z(4)) + 19*\log(3z_q(0)z_p(2)/z(3)) + 70*\log(z_p(2)/z(2)) \quad (20)$$

Substituting the values of z, z_q, and z_p from (19), we get

$$\begin{aligned} LogL = C &+ 30\log q + 11\log(q+z) + 100\log p \\ &+ 100\log(p+z) + 100\log(p+2z) - 100\log(1+z) \\ &- 100\log(1+2z) - 20\log(1+3z) - 11\log(1+4z) \\ &- 11\log(1+5z) \end{aligned} \quad (21)$$

where C is a constant.

Using MATLAB, the minimization of $LogL$ given in (21) is carried out. The value of C has no effect on the minimization process since it is a constant. The maximum likelihood estimates of p and z are found to be

$$\hat{p} = 0.822 \pm 0.028, \hat{z} = 0.521 \pm 0.178$$

The corresponding estimates of a and b were obtained as

$$\hat{a} = 1.29, \hat{b} = 0.3$$

Table 5. Simulation and Analytical Values of infected clusters for cluster size of 4

Infection Pattern $\{I_r\}$ $r = 0\ 1\ 2\ 3$	Observed number of clusters (Simulation)	Fitted values from the analytical model
1-1-1-1	5	3.1
1-1-2	6	4.1
1-2-1	19	14.8
1-3	70	67.3

Since b is less than unity, the beta-distribution is J-shaped with an infinite ordinate at p=1. Substituting the values of p, q, and z into the third column of Table 4, the fitted values for our chain-binomial model with variable p are calculated as seen in the third column of Table 5. The fitted values are the total number of clusters for the corresponding infection pattern chain given in the first column.

In the second scenario, we observe the scalability of the cluster based dynamic fanout scheme. We increase the cluster size to 5, which increases the total number of nodes in the system to 500. Similar to Table 4, the first column of Table 6 shows all the possible infection pattern chains for a cluster of size 5. The chain-binomial probabilities for all these chains are given in the second column of Table 6. The third column shows the expected probabilities of the chain-binomial model.

Table 6. Chain-binomial Probabilities for cluster size of 5

Infection Pattern $\{I_r\}$ $r = 0\ 1\ 2\ 3\ 4$	Expected number of Clusters (analysis)	Expected Values of Probabilities
1-1-1-1-1	$24nq^6p^4$	$24z_q(5)z_p(3)/z(9)$
1-1-1-2	$12nq^5p^4$	$12z_q(4)z_p(3)/z(8)$
1-1-2-1	$12nq^4p^3(1-q$	$12z_q(3)z_p(3)(1+q+12z)/z(8)$
1-1-3	$4q^3p^4$	$4z_q(2)z_p(3)/z(6)$
1-2-1-1	$12nq^4p^4(1-q$	$12z_q(3)z_p(3)(1+q+12z)/z(8)$
1-2-2	$6q^2p^2(1-q^2)^2$	$6z_q(1)z_p(3)[76z^2+(17+19q)z+(1+q)^2]/z(7)$
1-3-1	$4nqp^3(1-q^3)$	$4z_q(0)z_p(3)[38z^2+(12+9q)z+(1+q)^2]/z(7)$
1-4	p^4	$z_p(3)/z(3)$

From the third column of Table 6 and the simulation data in second column of Table 7, the log-likelihood function is given as follows:

$$\begin{aligned} Log\ L = C &+ 52\log q + 11\log(q+z) + 5\log(q+2z) \\ &+ 4\log(q+3z) + 3\log(q+4z) + \log(q+5z) \\ &+ 100\log p + 100\log(p+z) + 100\log(p+2z) \\ &+ 100\log(p+3z) + 1\log(1+q+12z) \\ &+ 1\log(76z^2 + (17+19q)z + (1+q)^2) \\ &+ 41\log(38z^2 + (12+9q)z + (1+q)^2) \\ &- 100\log(1+z) - 100\log(1+2z) - 100\log(1+3z) \\ &- 52\log(1+4z) - 52\log(1+5z) - 52\log(1+6z) \\ &- 51\log(1+7z) - 4\log(1+8z) - \log(1+9z) \end{aligned} \quad (22)$$

where C is a constant.

Using MATLAB, the minimization of $LogL$ was carried out. The maximum likelihood estimates of p and z are found to be

$$\hat{p} = 0.8444 \pm 0.0113, \hat{z} = 0.6122 \pm 0.0221$$

The corresponding estimates of a and b were obtained as

$$\hat{a} = 1.377, \hat{b} = 0.253$$

The shape of the beta-distribution is identical to the one obtained with cluster size 4, as the value of $\hat{b}$ is less than 1.

Table 7. Simulation and Analytical Values of the infected clusters for cluster size of 5

Infection Pattern $\{i_r\}$ r= 0 1 2 3 4	Observed number of clusters (Simulation)	Fitted values from the analytical model
1-1-1-1-1	1	0
1-1-1-2	1	0
1-1-2-1	1	1.5
1-1-3	1	0.65
1-2-1-1	1	1.5
1-2-2	6	10.3
1-3-1	41	37.72
1-4	48	51.24

6. CONCLUSIONS

We have described two protocols for information dissemination in large-scale communication systems by using dynamic epidemic protocols. For the Round-Based dynamic fanout protocol, the number of infected processes per round is based on the frequency distribution provided by users. This helps users quantify fanout to control the epidemic infection and allows users to fine tune this parameter. For the Cluster-Based dynamic fanout protocol, the fanout is not constant among the various clusters, unlike the round-based dynamic fanout approach. We have observed the infection pattern among the various clusters during every round. Simulation results for cluster size four and five have proved that the infection pattern closely follows a beta distribution.

Following are the issues that need further investigation: i) For the Round-Based dynamic fanout approach, the design of a good frequency distribution of infected nodes provided by the system user needs to be carefully examined. ii) The Cluster-Based dynamic fanout approach needs to be implemented on a large cluster size. We aim to address these issues in our future work.

REFERENCES

[1] Bailey, N. 1975. *The Mathematical Theory of Infectious Diseases and its applications*, Hanfer Press.

[2] Birman, K., Hayden, M., Ozkasap, O., Xiao, Z., Budiu, M., and Minsky, Y. 1999. Bimodal multicast. *ACM Transactions on Computer System*.

[3] Daley, D., and Gani, J. 1999. *Epidemic Modelling. An Introduction*, Cambridge Express.

[4] Demers, A., Greene, D., Hauser, C., Irish, W., and Larson, J. 1987. Epidemic algorithms for replicated database maintenance. In *Proc. of the Sixth Annual ACM Symposium on Principles of Distributed Computing*.

[5] Eugster, P., Guerraoui, R., Handurukande, S., Kermarrec, A., and Kouznetsov, P. 2003. Lightweight probabilistic broadcast. *ACM Transaction on Computer Systems*

[6] Gupta, I., Birman, K., and Van Renesse, R. 2002. Fighting fire with fire: using randomized gossip to combat stochastic scalability limits. *Quality and Reliability Engineering International*.

[7] Gupta, I., Van Renesse, R., and Birman, K. 2001. Scalable fault-tolerant aggregation in large process groups. *Dependable Systems and Networks*.

[8] Hedetniemi, S., Hedetniemi, S., and Liestman, A. 1988. A survey of gossiping and broadcasting in communication networks. *Networks.*

[9] Jelasity, M., Montresor, A., and Babaoglu, O. 2004. Detection and removal of malicious peers in gossip-based protocols. In FuDiCo II:S.O.S. Bertinoro, Italy. http://www.cs.utexas.edu/users/lorenzo/sos.

[10] Jelasity, M., Montresor, A., and Babaoglu, O. 2005. Gossip-based Aggregation in Large Dynamic Networks. *ACM Transactions on Computer Systems*.

[11] Jelasity, M., Kowalczyk, W., and Van Steen M. 2003. Newscast computing. Technical Report IR-CS-006, Department of Computer Science, Amsterdam, The Netherlands.

[12] Karp, R., Schindelhauer, C., Shenker, S., and Vocking, B. 2000. Randomized rumor spreading. *IEEE Symposium on Foundations of Computer Science*.

[13] Kempe, D., Kleinberg, J., and Demers, A. 2001. Spatial gossip and resource location protocols. In *ACM Symposium on Theory of Computing*.

[14] Kermarrec, L. Massouli´e, and Ganesh, A. 2003. Probabilistic reliable dissemination in large-scale systems. *IEEE Transactions on Parallel and Distributed Systems*.

[15] Luo, J., Eugster, P., and Hubaux, J. 2003. Route driven gossip: Probabilistic reliable multicast in ad hoc networks. *Infocom.*

[16] Peersim: http://peersim.sourceforge.net

[17] Van Renesse, R., Minsky, Y., and Hayden, M. 1998. A gossip-style failure detection service. In *IFIP Conf. on Distributed Systems Platforms an Open Distributed Processing*

[18] Van Renesse, R., Birman, K., and Vogels, W. 2003. Astrolabe: A robust and scalable technology for distributed systems monitoring, management, and data mining. *ACM Transactions on Computer Systems.*