Adaptive Gravitational Gossip: A Gossip-Based Communication Protocol with User-Selectable Rates

Kenneth Hopkinson, Member, IEEE, Kate Jenkins, Kenneth Birman, James Thorp, Fellow, IEEE, Gregory Toussaint, Senior Member, IEEE, and Manu Parashar, Member, IEEE

Abstract— Gossip-based communication protocols are attractive in cases where absolute delivery guarantees are not required due to their scalability, low overhead, and probabilistically high reliability. In earlier work, a gossip-based protocol known as gravitational gossip was created that allows the selection of quality ratings within subgroups based on workload and information update frequency. This paper presents an improved protocol that adds an adaptive component that matches the actual subgroup communication rates with desired rates coping with network variations by modifying underlying gossip weights. The protocol is designed for use in environments where many information streams are being generated and interest levels vary between nodes in the system. The gossip-based protocol is able to allow subscribers to reduce their expected workload in return for a reduced information rate. The protocol is a good fit for applications such as military information systems, sensor networks, and rescue operations. Experiments were conducted in order to compare the merits of different adaptation mechanisms. Experimental results show promise for this approach.

Index Terms— Adaptive Communication, Epidemic Protocols, Publish/Subscribe Systems

1 Introduction

THIS article introduces adaptive gravitational gossip (AGG), a group communication protocol that allows subgroups in the system to select transmission quality levels, which are maintained with high probability. The protocol uses heuristics to uphold information dissemination targets despite changes in network conditions. In a previous workshop article [1], a static version of the gravitational gossip (GG) protocol was introduced based on a mathematical model of gossip performance. The AGG protocol builds on this earlier work to allow gossip probabilities to adapt over time based on changing conditions. Experimental results show that the AGG protocol has great promise for use in situations where subgroups have different information targets, network conditions are unstable, and scalability is an important consideration.

The essence of gossip protocols is that nodes communicate by randomly and unreliably sharing information. Gossip protocols are also known as epidemic protocols because the spread of information mimics that of a disease. Each node periodically chooses another at random and sends a subset of the information that it has. No data is kept regarding which nodes have which information, which yields a relatively stateless protocol compared to traditional reliable group communication. Messages are sent unreliably using UDP. The process is shown in Fig. 1. The process is only probabilistically reliable, but is powerful in practice. Reliable group communication protocols must ensure that all nodes receive all messages, and this becomes increasingly difficult as group sizes grow. In large networks, it is likely that some node is having a connectivity or congestion problem at any given time, even if the identities of the problem nodes vary. Gossip allows nodes not experiencing difficulties to receive all information with high probability. Intermittently problematic nodes get as much information as they can. Gossip also spreads the work to publish information across the group's membership, leading to better scalability.

The static GG protocol takes gossip further. In many settings, different members of a group will have different interest levels in information. One can imagine a situation where interest in an information stream decreases with distance, as in Fig. 2. Nodes closer to the information source would like a higher percentage of periodic updates than those farther away. Nodes receiving less information should do less work to maintain the overall system.

The use of adaptive weights in epidemic protocols is relatively new. Two notable instances of adaptive epidemic protocols focused on choosing the minimum gossip rate such that nearly all members receive information

- *K. Hopkinson is with the Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433 E-mail: kenneth.hopkinson@afit.edu.*
- *K. Jenkins is with Akamai Technologies, Cambridge, MA 02142 E-mail: kjenkins@akamai.com.*
- *K. Birman is with the Department of Computer Science, Cornell University, Ithaca, NY 14850 E-mail: ken@cs.cornell.edu.*
- *J. Thorp is with the Department of Electrical and Computer Engineering, Virginia Polytechnic Institute, Blacksburg, VA 45445 E-mail:: jsthorp@vt.edu.*
- *G. Toussaint is with the Department of Electrical and Computer Engineering, Air Force Institute of Technology. Wright-Patterson AFB, OH 45433 E-mail: Gregory.Toussaint@afit.edu.*
- *M. Parashar is a consultant for the Electric Power Group, Pasadena, CA 91101 E-mail: parashar@electricpowergroup.com.*

Manuscript received May 28, 2007. *The views expressed in this document are those of the authors and do not reflect the official policy or position of the United States Air Foirce, Department of Defense, or the U.S. Government.*

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **MAY 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **Adaptive Gravitational Gossip: A Gossip-Based Communication Protocol with User-Selectable Rates**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Cornell University,Department of Computer Science,Ithaca,NY,14853**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Gossip-based communication protocols are attractive in cases where absolute delivery guarantees are not required due to their scalability, low overhead, and probabilistically high reliability. In earlier work, a gossip-based protocol known as gravitational gossip was created that allows the selection of quality ratings within subgroups based on workload and information update frequency. This paper presents an improved protocol that adds an adaptive component that matches the actual subgroup communication rates with desired rates coping with network variations by modifying underlying gossip weights. The protocol is designed for use in environments where many information streams are being generated and interest levels vary between nodes in the system. The gossip-based protocol is able to allow subscribers to reduce their expected workload in return for a reduced information rate. The protocol is a good fit for applications such as military information systems, sensor networks, and rescue operations. Experiments were conducted in order to compare the merits of different adaptation mechanisms. Experimental results show promise for this approach.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**14**	

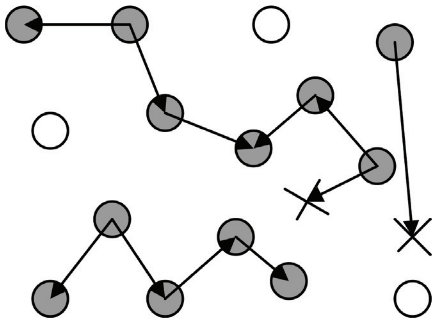

Fig. 1. In gossip, unreliable packets are sent to random destinations from nodes with information during each round. Nodes without information are light while nodes with the information are dark. Arrows represent gossip messages. An X represents a lost packet.

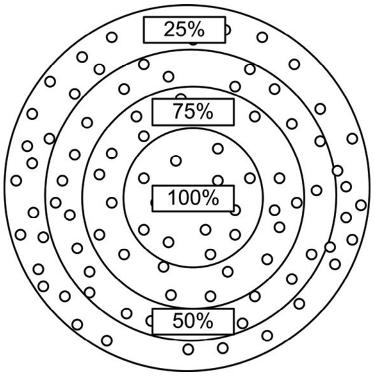

Fig. 2. In illustration of gravitational gossip: interest in information decreases with distance

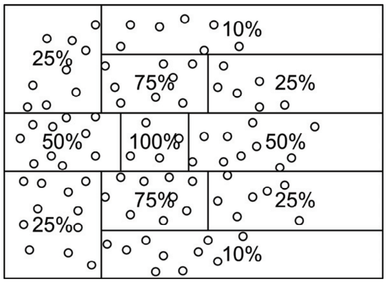

Fig. 3. A more typical use of the protocol. Interest levels may not correspond directly to distance. Also, nodes are clustered based on both interest and proximity to make adaptation easier.

based on factors like network conditions and buffer sizes [2] [3]. AGG goes beyond prior work by being the first epidemic protocol to allow multiple subgroups to each specify desired information rates, and to adapt gossip settings to meet those rate targets as network conditions change. Because conditions tend to be similar in a local area, it is possible to have subgroups with similar target information rates, but in different locations, as in Fig. 3.

The AGG protocol has many applications. These uses tend to fit a pattern where periodic information updates are broadcast, where interest in an information stream decreases by some criterion, often distance, and reasonable actions can be taken with a subset of the published information. There is also a tendency to have numerous simultaneous information streams, making it impractical to fully subscribe to them all. For scalability reasons, it becomes important to provide the following abilities.

1) Allow nodes to only subscribe to a fraction of the information that is being sent in a given stream.

2) Ensure that a publisher of information will have to do as little work as possible as the system grows.

3) Receivers will do their fair share of work, but will do proportionately less if they receive less information.

A potential application for the AGG protocol is information-centric military operations. The military is moving towards a more information-rich environment for its warfighters. This trend appears in the Department of Defense's Joint Vision 2020 [4] and in the addition of Cyberspace to the Air Force mission statement [5] [6]. Efforts to better manage large numbers of publish-subscribe information streams in military communication, like the JBI information system [7], point to a future where trade-offs will be required in the information flow across bandwidth-constrained battlefronts. Tools like AGG will allow a richer and more adaptive set of "dialable" information settings, which will allow more constrained sections of the network to "dial" their expected update rate down while others "dial" rates up. Achieving similar results using traditional protocols effectively means creating a publish-subscribe group for each subgroup with a different target. This solution has at least as great an overhead on the sender (i.e. publisher of information) as traditional reliable multicast protocols. AGG, by contrast, spreads the load of sending updates to all of the system's nodes. Members receiving fewer updates do less work in terms of messages sent and received. Similar requirements exist in classes of sensor networks and rescue operations.

The article begins by describing the basic GG protocol. A description of the new adaptive addition to the protocol follows. Experimental results are presented to show how the new protocol performs in control-based scenarios. A summary and a set of major conclusions follow.

3 Static Gravitational Gossip

3.1 Gravitational Gossip Mathematical Model

This section reviews the mathematical model, which forms the basis of the static gravitational gossip (GG) protocol, first introduced in [1]. The GG protocol is based on an epidemic model incorporating variable subpopulation infectivity and susceptibility rates developed by Busenberg and Castillo-Chavez in 1991 [8]. In traditional epidemic protocols, nodes choose others to send gossip messages to using a uniform probability distribution. The potential receivers of a gossip are the members of a process group, which is a set of nodes that subscribe to receive such information. An intuitive way to envision this process is that when it is time for a node within an epidemic-based process group of size N to gossip to another node in the system, it rolls an N-sided die to choose which other node to send a gossip message to. This die is fair, which means that no region of the die is more likely than any other. GG weights the gossip die so that some sides are more likely to appear than others.

Table 1 summarizes GG's major parameters. Nodes in a process group are divided into subgroups. Each subgroup is associated with a rating $r \in [0,1]$ that equates to

TABLE I
UNITS IN GRAVITATIONAL GOSSIP

Symbol	Type	Meaning
r	Set by User	The target information rate for a subgroup in the range [0, 1], which equates to 0% to 100%.
I_r	Set by User	The infectivity (tendency to gossip) for a subgroup with target information rate r. In this article, $I_r = r$ in all cases.
S_r	Controlled Parameter (set by algorithm)	The susceptibility (tendency to be gossiped to) for a subgroup with target r. Also called *gossip weight.*
gossip weight	Controlled Parameter (set by algorithm)	Synonymous with S_r.
timeout	Set by User	The amount of time a message will be gossiped in the system before it expires.
T	Equation Variable	A time value, not necessarily equal to *timeout*, which appears in some equations.
δ	Set by User	How much error can be tolerated for subgroups with a rate of 100%. (i.e. δ of 0.01 would be a target rate of 99.9%.)
t	Equation Variable	Time used in gossip equations, expressed in rounds (i.e. $t+1$ is one round higher than t).
$x_r(t)$	Equation Parameter	Percentage of subgroup members at rating r uninfected at time t.
N_r	Set by User	Number of nodes in the subgroup with rating r.
num_ nodes	Controlled Parameter (set by algorithm)	Number of nodes to gossip to per round from a given node.
message_ percent	Controlled Parameter (set by algorithm)	Percent of available messages included per gossip transmission.
quality contribution	Equation Parameter	Source infectivity times Destination Susceptibility.
$OI(S_k)$	Controlled Parameter (set by algorithm)	The observed infectivity (rate of information receipt) when a susceptibility of S_k is used.
I_1	Equation Parameter	A subgroup with a target information rate of 100%.
γ	Controlled Parameter (set by algorithm)	Used in a binary search that scales $x_r(t)$ to expire at *timeout.*

the target rate for information updates. A higher rate yields more information, but requires more work from a subgroup's members in terms of the expected number of messages that will be sent and received. Although non-uniform distributions in epidemic protocols have been discussed for over 20 years in the literature, including an early article by Demers [9], GG's innovation is that it uses a mathematical model of gossip behavior to allow users to intelligently choose the gossip probabilities, also called gossip weights and susceptibility values in this article, to achieve the desired expected distribution of information by a given expiration time, called a timeout value.

Borrowing from epidemic terminology, each subgroup of nodes in a process group has two parameters, its infectivity (I) and susceptibility (S). A node that has not received a piece of information is said to be susceptible. Once that node has the information, it is infected and able to pass the new data along to others.

To be more precise, we have the following definitions.

Define I_r = the infectivity of a subgroup with a target information rate *r*, which corresponds to the probability that a member of the subgroup with knowledge of a data item will gossip to other group nodes. $I_r \in [0,1]$.

Define S_r = the susceptibility of a subgroup with target rate *r*, which is the probability that a subgroup member will be targeted with a gossip message. $S_r \in [0,1]$.

The infectivity and susceptibility rates of two nodes i and j multiplied together, $I_{r_i} \cdot S_{r_j}$, gives the probability that node i will send a gossip message to node j, where r_i and r_j are the two nodes' subgroup ratings.

Because a publisher (aka sender) of information will always have 100% of the information that it generates, an information publisher will always be a part of subgroup 1 (1 signifying 100%). The special terms I_1 and S_1 refer to the infectivity and susceptibility, respectively, for this subgroup. While multiple publishers can be added with a simple extension to the formulas presented here, we will concentrate on the single publisher case in this article in order to make the presentation cleaner.

In GG, a recurrence relation was found to determine the probability that a node would not be infected after t gossip rounds given fixed values of I and S.

The lifetime of a gossip message is expressed in terms of a number of gossip rounds because nodes choose to gossip to others in their group periodically, although node clocks are not assumed to be synchronized to the point where a round of gossip occurs at the same time at all group nodes. In this way, the lifetime of a message can be thought of as a discrete value in terms of the number of gossip rounds remaining.

Define N_r = the number of nodes with rating r, and I_r, S_r respectively as the infectivities and susceptibilities of nodes with rating r.

Define $x_r(t)$ = probability node m with rating r is still susceptible at time t

Then,

$$x_r(t+1) = x_r(t) \cdot$$
$$P(\text{each machine that gossiped to rating r at time t + 1 missed m})$$

$$x_r(t+1) = x_r(t) \cdot (1 - 1/N_r)^{\text{num gossips to rating } r \text{ at time t+1}}$$

$$x_r(t+1) \approx x_r(t) \cdot (1 - 1/N_r)^{N_r \cdot S_r \cdot \sum_j I_j \cdot N_j \cdot (1 - x_j(t))}$$

By basic calculus [10], $(1 - 1/x)^{-x} = e$ as x → ∞, so

$$x_r(t+1) \approx x_r(t) \cdot e^{-S_r \cdot \sum_j I_j N_j \cdot (1 - x_j(t))}$$

where the sum of the exponents is calculated over all of the different ratings. For purposes of approximation, it is assumed that the random variable in the exponent is equal to its expected value.

This equation can be broken down into a series of recurrence relations for each of the system's subgroups.

$$
\begin{aligned}
&x_1(t+1) \approx x_1(t) \cdot e^{-S_1 \cdot \sum_j I_j \cdot N_j \cdot (1 - x_j(t))} \\
&x_{r_1}(t+1) \approx x_{r_1}(t) \cdot e^{-S_{r_1} \cdot \sum_j I_j \cdot N_j \cdot (1 - x_j(t))} \\
&x_{r_2}(t+1) \approx x_{r_2}(t) \cdot e^{-S_{r_2} \cdot \sum_j I_j \cdot N_j \cdot (1 - x_j(t))} \\
&\quad \ldots \\
&x_{r_k}(t+1) \approx x_{r_k}(t) \cdot e^{-S_{r_k} \cdot \sum_j I_j \cdot N_j \cdot (1 - x_j(t))}
\end{aligned}
$$

Given these equations, the fraction of infected nodes for the subgroup at rating r in round t is $1 - x_r(t)$. An important property was also derived that follows from induction on the recurrence relations above. Namely,

$$\forall t, x_r(t) \approx (x_1(t))^{S_r / S_1}$$

This follows by induction since

$$
\begin{aligned}
x_r(t+1) &= x_r(t) \cdot e^{-S_r \cdot \sum_r I_j N_j \cdot (1 - x_j(t))} \\
&= (x_1(t))^{S_r/S_1} \cdot (e^{-S_r \cdot \sum_r I_j N_j \cdot (1 - x_j(t))})^{S_r/S_1} \\
&= (x_1(t) \cdot e^{-S_r \cdot \sum_r I_j N_j \cdot (1 - x_j(t))})^{S_r/S_1} \\
&= (x_1(t+1))^{S_r/S_1} \text{ and } x_r(0) \approx x_1(0)^{S_r/S_1} \text{ for } N_1 \text{ large}
\end{aligned}
$$

Each node within a given subgroup has the same desired rate of information, *r*. Because *r* is a real number, it is possible that many or most nodes will be in subgroups by themselves unless care is taken to make rating discrete and to cluster nodes with close desired rates together. The derivation above depends on the law of large numbers. A setting with many singleton subgroups serves as a worst-case for the gravitational gossip protocol.

3.2 Gravitational Gossip Parameter Prediction

The equations derived in the previous section give a method for determining the number of infected nodes at rating r at round t given gossip parameters I_r and S_r. The inverse problem is often the one that is of most interest. Given a set of target subgroup ratings, can the gossip parameters that will result in nodes receiving information at the prescribed rates be determined mathematically? The equations derived in the previous section are in the form of recurrence relations that do not lend themselves to a straightforward inversion. This section briefly reviews the static algorithm [1] for finding gossip parameters that can be used to achieve the target rating in expectation for each subgroup.

Mathematically, the goal of the inverse algorithm is to find values $S_1, S_{r_1}, ..., S_{r_k}$ given the target infectivity values $I_1, I_{r_1}, ..., I_{r_k}$ input by the users such that:

$$
\begin{aligned}
&x_1(timeout) \approx 0 \\
&x_{r_1}(timeout) \approx 1 - r_1 \\
&\quad \ldots \\
&x_{r_k}(timeout) \approx 1 - r_k
\end{aligned}
$$

and $I_{r_1} / I_1 = r_1, ..., I_{r_k} / I_1 = r_k$ so we will meet the goal that a machine at rating *r* will initiate *r* times as many gossips as a machine at rating 1, and a machine at rating *r* will have a probability *r* of receiving a message before it times out. I_1 always includes the publisher, which has value 1.0 since it will have 100% of the information it generates.

The parameter prediction algorithm works as follows:

For any fixed parameter setting, there is some smallest round number T such that $x_1(T) < \delta$ where δ is an arbitrary value asking for how closely the expected value of the first subgroup, which contains the sender (i.e. publisher), should be to 1. For example, a δ value of 0.01 would signify that the expected rate of information will be 99.9% of all published information in the first subgroup of the system.

Given the previous section's equations, the fraction of infected nodes for the subgroup at rating r in round T is $x_r(T) \approx (x_1(T))^{S_r/S_1} < \delta^{S_r/S_1}$. If we let $S_1 = 1$ and pick S_r such that $x_r(T) < 1 - r$ then this means that S_r should be chosen so that $\delta^{S_r} = 1 - r$. Solving for S_r yields:

$$S_r = \log_\delta(1 - r) = \log_2(1 - r) / \log_2(\delta), \forall r$$

This equation gives the desired fraction of infected nodes at each rating r at round T, but we want this to be achieved by round *timeout*. To achieve this, we can globally scale the susceptibilities by some value, which we will call γ. No matter what the value of γ is, whenever $x_1(T) < \delta$ the fraction of susceptibles at the other ratings will satisfy the desired proportions. Varying γ just affects the time when this occurs. Thus, we binary search to find a value of γ for which $x_1(timeout) < \delta$, but $x_1(timeout - 1) \geq \delta$. We test a given γ by running the recurrence relations with the corresponding parameters.

It should be noted that the term time, in a slight abuse of notation, is used here to refer to a number of rounds, which are discrete, rather than a continuous value. Time is used in a similar manner in the rest of the article.

4 Adaptive Gravitational Gossip

The static GG parameter prediction method works well in stable network conditions, but many real-world applications run over shared networks with unpredictable levels of competing traffic, network congestion, and sporadic connectivity. The static protocol works best in environments with large numbers of members, high timeout values, low network communication errors, and low message latencies. Important applications, such as the wide-area monitoring and control situations outlined in Section 2, require a protocol that can adapt as conditions change. While other adaptive gossip protocols exist that change their gossip distribution based on group member information in response to user queries [11, 12] and physical node locations [13, 14], the protocol in this article is able to allow the creation of subgroups, each with their own gossip rate targets, and can adapt the overall system as network conditions change.

Each node keeps a record of the percentage of messages received over time. This can be done because messages are assumed to be periodic and the periodicity is known to all members of the group, as mentioned in Section 4.

This section begins by describing a protocol framework to support adaptive behavior. The protocol's central premise is that if information is known regarding the process group and subgroup membership, the infectivities (the rate of information desired), and the publication rate from the group's sender, then appropriate susceptibilities (aka gossip weights) can be found such that subgroups desiring more information will gossip more to maintain the overall information flow within the group.

The sender must publish new information at a predictable rate in the adaptive GG protocol in order for the group members to calculate the quality rate of information that they are receiving. The rate that the sender publishes new information is assumed to be periodic in this article. The goal is to ensure that the desired rate of information for each subgroup (its infectivity) is equal to the received rate. (I.e. each node in a subgroup with an infectivity of 0.25 should receive 25% of all published information, on average, by the time it expires.)

4.1 A Basic Adaptive Protocol Framework

The previous sections beg the question of how adaptive information is gathered into and disseminated out from adaptive weight adjustment heuristics. A basic protocol framework has been created in order to illustrate one way of implementing such a protocol and to allow experiments to compare three competing adaptive heuristics.

Static gossip weight calculations are triggered whenever a new group is created, when the set of the group's subgroups changes, or when membership changes occur. In the description of the group communication protocol that follows, for each node that belongs to a group, it will be a member of exactly one subgroup within that group. In publish-subscribe terminology, all nodes in the group would be considered subscribers to an information stream. In the terminology used in this article, the senders of information to a group are the group's publishers. It is important to keep in mind that all members in the group are likely to periodically forward information in the form of gossip, but only the senders (i.e. publishers) are sources of new information to the group.

In the basic adaptive protocol developed in this article, static weight calculations follow the mathematical model outlined in Section 3. The infectivity is equal to the desired rate of information for each subgroup, where the value ranges from 0 (no information) to 1 (100% of the published information). The susceptibility weight calculations for each subgroup are performed by the sender within a group. (As mentioned previously, a single sender is assumed in this article for clarity of presentation. The same technique is easily extended to multiple senders by keeping each sender in its own subgroup.) A system invariant is that the group's sender belongs to its own subgroup. This is necessary due to the fact that the senders inherently possess 100% of all messages that they send. So, for example, if a sender were placed in a subgroup with a susceptibility of 0.20 and four other nodes then the target information rate would automatically be satisfied since 100%/5 nodes = 20%/node. For this reason, the sender will be in its own subgroup, and the four remaining nodes will be placed in a separate subgroup.

As the protocol progresses, each node keeps a record of the percentage of messages received over time. This is possible because messages are assumed to be periodic with a known periodicity, as mentioned in Section 4.

4.1.1 Overview of Adaptive Protocol Messages

All messages in the basic protocol framework are sent unreliably using UDP packets. The purpose and content of each message type is described in this subsection.

4.1.1.1 Gossip Messages

Every group member periodically sends gossip messages. Gossip intervals are unsynchronized between group members. Gossip cycles occur at roughly the same frequency at every group node, but the clocks and gossip trigger times vary from node to node. The gossip frequency is a group-wide parameter in the system.

In the group communication system, each new piece of information published to the group is called a gossip fragment. Gossip fragments expire and are no longer shared after a fixed time set on a group-wide basis.

A node will gossip to another node at random according to the gossiping node's infectivity and the selected recipient's susceptibility. Initial calculations for the infectivity and susceptibility follow the method outlined in Section 3. From then on, infectivity and susceptibility values are adapted to compensate for changing network conditions using heuristics such as those in Section 5.

When a gossip message is sent, the message consists of a series of gossip fragments appended to each other. The number of fragments is dictated by the formula in Section 4.2. Fragments are chosen at random from the sending nodes' set of known unexpired published information.

4.1.1.2 Periodic Gossip Weight Recalculation

Denote the number of rounds before a message expires as T . In the adaptive gossip protocol, each group member is told how frequently each sender (i.e. publisher) sends new information to the group. The members also know how long messages should be propagated before they expire and are no longer of interest. Given this information, group members are able to calculate the percentage of information that they are receiving based on the messages received in a given time period. For example, in the experiments run in Section 6, the percentage of messages received was calculated over a timeframe of T . In these experiments, 20 messages were published per 100 ms round and messages expired after 20 rounds. Therefore, simple division of the messages received by the lifetime of each message gives the quality measurement over this time period. For example, if an average of 5 messages were received per round then 5 messages / 20 messages published per round = a 25% quality rating. This result can be aggregated with the others across the T timeframe to get an overall average. So, if T is 4 then we can use the average quality rating for each of the 400 100ms rounds between time δ and δ + 4. Once this value is calculated, information can be gathered from each subgroup to see if the average member is receiving the desired quality rate.

Every $2 \cdot T$ rounds, the message sender randomly chooses one node in each subgroup and sends a gossip quality request. $2 \cdot T$ is chosen to allow ample time for new gossip weights to propagate through the system before taking performance measurements. When a node receives a subgroup quality request, it relays a gossip message to each of the other members of its subgroup. The subgroup nodes' responses are totaled and averaged before they are returned to the sender. Weight adjustments are performed for each responding subgroup using heuristics such as those in Section 5. All messages are sent unreliably. If a response is not received then the last known value is used. Weights readjustment results are distributed in unreliable gossip messages sent to one randomly chosen member of each subgroup. When a subgroup member receives a weight readjustment value, it sends an unreliable copy to all other subgroup members.

4.1.1.3 Recovering When Weight Updates Do Not Arrive at Group Nodes

The gossip weight update messages described in the previous subsection are sent unreliably so a mechanism is needed to allow nodes to receive the new values when an original update message is lost. To do so, the latest gossip weights and their timestamp are hashed together into a 32-bit value. The hash values are included with all gossip messages throughout the information dissemination process. In each gossip message, hash values are included for each group/sender with at least one information fragment included in the message. When a node receives a gossip message, the hash value(s) in the message are compared against a hash of the node's current gossip weight values. If the two hashes are not a match then a repair message is generated from the receiver to the source of the gossip message. Upon receipt of a repair message, if the receiving node has newer weights than those contained in the message then those newer values are returned in a message. Once again, all messages use unreliable UDP packets. Message losses end the exchange and no attempt is made to detect or recover from them.

4.1.1.4 Potential Protocol Enhancements

This protocol is relatively simple. It could be strengthened in a number of ways. Large subgroups, subgroups with diverse topologies, or subgroups with varied communications characteristics could benefit from a hierarchical communications infrastructure or an automated mechanism to split nodes into smaller subgroups and then recombine when appropriate. The adaptive protocol centralizes all subgroup gossip weight calculations at group senders, but heuristic other than the initial adaptive do not require this. A representative member of each subgroup could do the job instead. To make this work, some type of leader election algorithm would be required.

4.2 Converting Gossip Weight to Transmission Probabilities

The gossip weight calculations described in Section 3, and the adaptive heuristics described in Section 5, use a mathematical model of gossip behavior and heuristic algorithms to find gossip weights for the subgroups affiliated with a group sender/publisher. Based on the terminology in Section 3, the goal of these calculations is to predict the susceptibility (gossip probability) value for the members of a subgroup in order to maintain the desired infection rate (percentage of published information received).

4.2.1 Calculating the Quality Contribution Factor

In the gossip algorithm, each node possesses an infectivity rating, which is the desired percentage of information received. To make an analogy with the spread of a disease, an individual's infectivity is their tendency to spread disease. An individual's susceptibility is their tendency to get ill around an infected individual. Similarly, a higher susceptibility in gossip means a node is more likely to be gossiped to, and a high infectivity means a node is more likely to send gossip to other individuals. (A goal of the protocol is for members with a higher desired information rate to do more work to maintain the group.)

The source infectivity multiplied by a destination subgroup susceptibility yields a raw value called the *quality contribution* from a given node to a given subgroup.

4.2.2 Dividing Quality Contribution Factors into Gossip Probabilities and Message Sizes

For a given quality contribution value, a trade-off needs to be made between the probability of gossiping to a node versus the amount of information contained in each gossip message. (A quality contribution of 0.5 can be achieved by sending half of known information updates to one node, a quarter to two nodes, etc.)

If the number of nodes gossiped to is minimized then the knowledge variance may be high from one subgroup member to another. At the same time, it is not desirable to send too many network messages. We have adopted the convention that the quality contribution should be divided so that the number of nodes to gossip to in a subgroup, called *num_nodes* here, is calculated as follows

$$num\_nodes = \left\lceil \frac{quality\_contribution}{target\_subgroup\_rating} \right\rceil$$

and that the percentage of messages included in each packet sent, called *message_percent* here, are equal to

$$message\_percent = \frac{quality\_contribution}{num\_nodes}$$

For example, if subgroup A has target infectivity 0.75 and susceptibility 0.25, and subgroup B has infectivity 0.5 and susceptibility 0.33 then for A->B, $num\_nodes = \lceil (0.75 \cdot 0.33)/0.5 \rceil = 1$ so each round one message will be sent from each node in subgroup A to a random member of subgroup B. $message\_percent = (0.75 \cdot 0.33)/1 = 0.495$ so each message will contain 49.5% of the known unexpired gossip fragments. The heuristic balances the number of gossip messages sent against the message sizes.

The heuristic for dividing a quality contribution into a probability of gossip versus the amount of information per gossip message worked well in the experiments run.

5 Adaptive Gossip Heuristics

One shortcoming of the gravitational gossip algorithm in Section 3 is that there is no closed-form solution. It is expressed as a recurrence relation. The inverse algorithm in Section 3.2 works well, but cannot take network losses into account. This section describes four adaptive heuristics, which adapt gossip weights over time to compensate for changes in network losses rates and other disruptions.

The four heuristics each run over the distributed framework described in Section 4. At periodic intervals, an adaptive algorithm will take in the average observed infectivity in each subgroup since the last time a new susceptibility value was computed. The adaptive algorithm computes new susceptibility values per subgroup based on the rate targets (subgroup infectivity values). These new susceptibility values are distributed throughout the communication group using a gossip-based mechanism and the cycle starts again. A description of each of the adaptive mechanisms that we tested is provided below.

5.1 A Linear Heuristic for Finding Gossip Weights

Finding a heuristic method to find appropriate gossip weights (i.e. susceptibilities) as network conditions change over time is a challenge. This section describes a simple linear heuristic, which begins with a numerical algorithm to solve the quality inversion problem in Section 3.2 in the face of changing network conditions.

To solve the inversion problem numerically, an algorithm must find the roots of the mathematical model equations in Section 3 based on a given set of quality rates. The problem is to find a set of susceptibility values S_r, for each subgroup with target rating *r,* such that the infectivity values of each subgroup I_r will be equal to their final rate of infection after the last round of gossip is over. Each message has an expiration time, which translates into a fixed number of gossip rounds over the message's lifetime, called *timeout*. The expected percentage of uninfected nodes (i.e. nodes without knowledge of a piece of information) by time *t* is given by the Section 3 equation

$$x_r(t+1) \approx x_r(t) \cdot e^{-S_r \sum_j I_j \cdot N_j \cdot (1-x_j(t))} \qquad (1)$$

$x_r(t)$ is the percentage of uninfected nodes at time t. S_r is the susceptibility (gossip weight) of the nodes in the subgroup with target information rate *r.* I_j is the infectivity for subgroup with target rate *j* (the rate *j* is the target percentage of nodes that will receive the message before it expires). N_j is the number of infected nodes in subgroup *j* by time *t*. The exponent is summed over all subgroups, which is denoted by the *j* subscript.

Equation (1) is a recurrence relation for finding the final percent of infected nodes given a starting infectivity value I_j over each subgroup with rating *j*. Matlab's lsqcurvefit method [15] is a non-linear root-finding algorithm in multiple dimensions. By specifying that susceptibility values are only valid over the range [0,1], lsqcurvefit can look for subgroup susceptibilities such that the actual information rates will equal those desired.

Equation (1) gives good approximations when the network has low congestion and low bit error rates, but can be inaccurate in other circumstances. Non-linear root-finding numerical methods like lsqcurvefit work by testing many evaluations of a user-supplied function to find a vector of values that result in an answer that is close enough (as defined by the user) to the true solution. A good set of susceptibility values under ideal network conditions can be found by supplying a function to evaluate equation (1) at time *timeout*. In real networks with shifting conditions, one could supply a function to lsqcurvefit that allowed a simulation or even a real network to operate long enough to be stable and then could use the resulting average information rate for each subgroup as the return value for that function evaluation. Unfortunately, lsqcurvefit and other similar methods can require hundreds of function evaluations before converging to a value. A long series of evaluations is slow and is also susceptible to changing network conditions in the middle of a function evaluation cycle. The effects of small changes in gossip weights are also likely to be lost in the noise caused by network fluctuations and sampling error.

A simple heuristic can capitalize on the Matlab-based non-linear least-squares inversion method by assuming a linear continuum between susceptibility weights. Pseudocode for this method is given below. All variables in the equation below are $\left[1 \cdot num\_nodes\right]$ vectors, and division and multiplication are to be performed term by term.

```
1. susceptibility ← least squares calculation
   (based on the target subgroup infectivity)
2. observed infectivity ← network protocol evaluation
3. modified susceptibility ← least squares calculation
   (observed infectivity based on calculated suceptibility)
4. true susceptibility ← susceptibility · $\frac{\text{susceptibility}}{\text{modified susceptibility}}$
```

This algorithm calculates susceptibility weights based on equation (1) (psuedocode line 1). Next, these weights are used under real network conditions for an evaluation period (4 seconds in the article's experiments) (line 2). Then, lsqcurvefit is used again to find the susceptibility values (modified susceptibility values in line 3) required in an ideal network to achieve the information rate that was actually observed in the network (observed infectivity in line 3). In step 4, we assume that there is a linear relationship between the input susceptibilities and output observed infectivities. We take the proportionate difference between the observed and desired infectivities and then scale the susceptibility values used by that factor.

This simple heuristic has been shown to produce an order of magnitude gain in accuracy when the initial susceptibility estimate is far from the target infectivity, but it is not a cure-all. Repeated applications can lead to divergence. Nonetheless, the experimental gains make this a good starting place before additional weight refinement.

5.2 Newton's Method

The Section 5.1 adaptive heuristic improves the accuracy of the initial susceptibility weights. The adaptive heuristic could be followed by the use of further n-dimensional non-linear root-finding iterations, but the number of function evaluations would be too high to be practical. A heuristic that works well is using separate instances of a modified form of Newton's Secant Method [16] in one-dimension independently for each of the subgroups. To briefly review, Newton's Method for one-dimensional root-finding is based on a truncated Taylor expansion around the function to be minimized, which searches for a minimum value x using an iterative approach. This search is successful when the starting value is close enough to the solution. The search uses equation

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

The Secant Method is a modification of Newton's Method, which uses difference equations in place of explicit derivatives. Hence, the last equation is transformed into:

$$x_{k+1} = x_k - f(x_k)\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

which converges superlinearly. Advanced root-finding algorithms exist, but a Secant method advantage is that it only depends on the previous function evaluation. This is a major strength as it helps shield the protocol from reliance on outdated protocol performance measurements.

The Secant method adapts gossip behavior by computing the next susceptibility values in order to minimize the difference between the desired information rate (infectivity value) and the observed rate (observed infectivity value). Periodically (every 4 seconds in the experiments), each subgroup uses the secant value to compute the new susceptibility for that subgroup to be used over the next time period. Subgroup *r* uses the information gathered from its members, using the process in Section 4.1, to find new susceptibility values using the equation

$$S_{k+1} = S_k - OI(S_k)\frac{S_k - S_{k-1}}{OI(S_k) - OI(S_{k-1})}$$

where S_k is the susceptibility value used in round *k* and $OI(S_k)$ is the average observed infectivity (information rate) that was achieved during the time period where susceptibility S_k was used. Two heuristics are applied to the Secant Method to improve its performance. First, no new calculation is invoked if the desired and observed information rates are within 0.02 of each other. Second, the returned susceptibility value is forced to be in the range [0.02,0.98) to ensure that it is within a sane range. These exact bounds are heuristics (i.e. [0.05, 0.95] might also work). Limits above 0 and below 1.0 are needed because it is difficult to measure the impact of a change in the input susceptibility on the output infectivity when the range includes the extreme ends 0 and 1.0.

There is no mathematical proof that this converges, but experimental evidence has shown that it does converge and does so rapidly under stable network conditions.

5.3 Proportional Controller

Newton's method works well in practice, but the resulting gossip weights (i.e., susceptibility values) can oscillate. Wild steps can also occur when network conditions remain relatively constant between measurement intervals making the delta between readings small. The outcomes measured will also drift naturally from one reading to the next. An alternative is to create a controller for the AGG protocol. As in Newton's method, one controller is used per subgroup to adapt its susceptibility values (i.e., gossip weights) in response to changing network conditions. Fig. 4 shows a simplified block diagram model of a network subgroup that can be used to understand how the controller can stabilize the gossip weights. The control model is only approximate because it treats the network as a linear system and models the nonlinearities as an additive disturbance. More advanced models are possible, but this simplified approach is sufficient to understand how the feedback mechanism functions and it has been effective in the experiments run. The meaning of the proportional controller's major components follows.

1) Reference (*r*)

The reference signal to the system, *r*, is equals to the subgroups' target information rate (i.e., infectivity rate).

2) Error (*e*)

The error, *e*, from each iteration is the difference between the target infectivity rate, *r*, and the observed infectivity rate in the network. The error signal is sent to the controller to create the next target susceptibility value (gossip weight) for the network subgroup.

3) Controller (*K*(*s*))

The controller, *K*(*s*), is modeled as a proportional controller that converts the error signal into a target susceptibility value. Proportional control is a relatively simple approach that allows the system to approximately track the reference signal while also rejecting the disturbances. For this effort, the controller was tuned to a constant value K_P, which was set to 75% of the maximum susceptibility value. The maximum susceptibility value is the susceptibility value for the subgroup when all subgroups desire a 100% information rate. At 100%, all subgroups receive all information with high probability. The susceptibility value was found using the Section 3 equations.

4) Plant Process (*P*(*s*)) and Disturbance (*d*)

The plant process, *P*(*s*), represents the behavior of the network modeled as a generalized linear system. The susceptibility (gossip weight) for the subgroup serves as an input to the network. As outlined in Section 4.1, the new subgroup susceptibility value will propagate throughout the group. Next, a user-defined amount of time will pass to allow periodic newly published gossip fragments to be gossiped through the group. After the user-defined time span, statistics are collected from the subgroup members, as described in Section 4.1, and the observed infectivity rate for the subgroup is used as the system output.

There are two observations about the network that guide the model development for the system. In general, it is not practical to give an exact model of network behavior without specifying detailed operating conditions. The first observation is that higher levels of network congestion lead to lower observed infectivity rates. It is difficult to model the relationship between congestion levels and infectivity rates, but is likely nonlinear. The second observation is that the gossip protocol operates in a network that cannot change its state instantaneously. That is, the network essentially has a memory of past states and

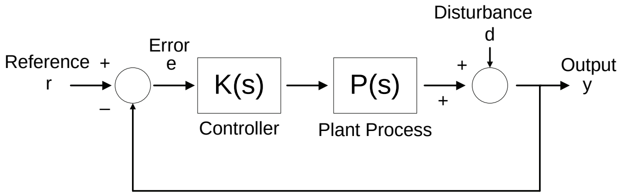

Fig. 4. Linear System Model with Disturbance

will not immediately move from one congestion level to another. These observations indicate that network model must account for the nonlinear processes and how the system responds to requested changes. A simple approach to accommodate these elements is to model the network with a first-order differential equation with additive disturbances. The first-order differential equation captures the memory features of the network and the additive disturbances represent the nonlinear behavior. In terms of the model, shown in Fig. 4, the first-order differential equation translates into the transfer function $P(s) = 1/(s+\beta)$, where β characterizes how quickly the system responds to changes. The disturbance, *d*, is added to the output of the network model to yield the system output, *y*. When a subgroup's gossip weight (susceptibility) is input into the network, the result is a percentage of nodes that will be infected on average. The result must be between 0 (0% infected) and 1 (100% infected).

5) Output (*y*)

The system output of the system, *y*, is the observed infectivity from the network. The output is determined through system measurements and used as a feedback signal to regulate the controller.

The frequency-domain (Laplace) analysis below derives the relationship between the output signal, the reference signal, and the disturbances as depicted in Fig. 4.

$$Y(s) = D(s) + P(s)K(s)[R(s) - Y(s)] \tag{2}$$

$$Y(s) = D(s) + P(s)K(s)R(s) - P(s)K(s)Y(s) \tag{3}$$

$$[1 + P(s)K(s)]Y(s) = D(s) + P(s)K(s)R(s) \tag{4}$$

$$Y(s) = \frac{D(s)}{1 + P(s)K(s)} + \frac{P(s)K(s)R(s)}{1 + P(s)K(s)} \tag{5}$$

Using a proportional controller and the linear system model described above, we have:

$$K(s) = K_P$$

$$P(s) = \frac{1}{s+\beta}$$

$$\begin{aligned} Y(s) &= \frac{D(s)}{1 + \frac{K_P}{s+\beta}} + \frac{\left(\frac{K_P}{s+\beta}\right)R(s)}{1 + \frac{K_P}{s+\beta}} \\ &= \frac{(s+\beta)D(s)}{s+\beta+K_P} + \frac{K_P R(s)}{s+\beta+K_P} \end{aligned}$$

If there is no disturbance, then the step response to a reference input signal is given by:

$$Y(s) = \frac{K_P}{s(s+\beta+K_P)}$$

and the corresponding output signal will be:

$$y(t) = \frac{K_P}{\beta+K_P}\left[1 - e^{-(\beta+K_P)t}\right]u(t)$$

The output signal will approach the fraction $K_P/(\beta+K_P)$, which, for $K_P >> \beta$, approaches 1 as a steady-state value. The experiments run always had $K_P >> \beta$, so the system would approximately track the reference signal.

If there is no reference signal and the disturbance input is a step function, then the output signal is:

$$Y(s) = \frac{s+\beta}{s(s+\beta+K_P)}$$

and the corresponding output signal will be:

$$y(t) = \left[\frac{\beta}{\beta+K_P} + \frac{K_P}{\beta+K_P}e^{-(\beta+K_P)t}\right]u(t)$$

The output signal will approach the fraction $\beta/(\beta+K_P)$, which, for $K_P >> \beta$, approaches 0 as a steady-state value. Therefore, the feedback system will attenuate the disturbance at the output.

In summary, if the proportional gain is large enough relative to the parameter β, then the output signal will closely track the reference signal and the disturbance will not play a significant role in the output. This control law is relatively simple and will yield reasonable results, but it can be improved by combining integral feedback with the proportional feedback.

5.4 Proportional-Integral Controller

The proportional controller does not have problems associated with derivative calculations that occur in Newton's method, but it does not converge as quickly (expected linear convergence versus superlinear convergence for Newton's Secant method). The proportional controller can be improved by including an integration term in the feedback control law to increase the performance in tracking a reference signal and in rejecting a disturbance. The new controller is called a proportional-integral controller and its corresponding transfer function is given by

$$K(s) = K_P + \frac{K_I}{s}$$

where K_P and K_I are constants that determine the relative weights of the proportional and integral components of the control signal. Substituting the new controller into the model in Fig. 4 and performing a similar analysis to the one in Section 5.2 yields the following output response:

$$Y(s) = \frac{s(s+\beta)D(s) + (K_P s + K_I)R(s)}{s^2 + (\beta+K_P)s + K_I}$$

With no disturbances, the system response to a step input as the reference signal is

$$Y(s) = \frac{K_P s + K_I}{s\left[s^2 + (\beta+K_P)s + K_I\right]}$$

Using the Laplace transform final value theorem, it can be shown that as $t \to \infty$, $y(t) \to 1$, which indicates the system eventually tracks the reference signal without error.

If only a disturbance signal is applied to the system as a step function and there is no reference signal, then the system response is given by

$$Y(s) = \frac{s+\beta}{s^2 + (\beta + K_P)s + K_I}$$

Again using the final value theorem, the output signal approaches zero as $t \to \infty$, so the system will completely reject disturbances given enough time to compensate for initial errors. This article's test cases demonstrate the performance advantages of the proportional-integral control law compared to the simpler proportional controller.

In this article's experiments, the values for the proportional and integral controller gains were found through experiments guided by Zeigler-Nichol's tuning rules [17].

The above controller design considered the model as a continuous-time system and used Laplace transform techniques in the analysis. The continuous-time representation is convenient and efficient for analysis, but the system will be simulated in a discrete-time environment. The general results are valid for a discrete-time implementation, but the details of the analysis and controller design are slightly different. The experiments run accounted for these differences and ensured the final system design maintained stability. An introduction to discrete-time control and the associated *z*-transform can be found in Ogata's book, "Discrete-Time Control Systems" [18].

6 Experimental Results

This section has four subsections. The first subsection describes the experimental setup. The second gives a description of the adaptive tests. The third presents experiments that compare the behavior of the adaptive methods presented in Section 5 using trials adapted from standard transient response tests. The fourth subsection compares the performance of the adaptive methods against TCP and UDP in the same set of transient response tests to illustrate the adaptive algorithm's strengths.

6.1 Experimental Setup

Standard control tests were duplicated to compare the effectiveness of the different adaptive algorithms. Detailed descriptions of the impulse, ramp response, and sustained oscillation tests used can be found in introductory control theory texts including Ogata's book, "Modern Control Engineering" [17]. The tests' purpose is to measure the stability and response times of adaptive heuristics in the face of changing input signals. Each test is briefly described in Section 6.2 along with its translation from the signal domain to a network communication environment.

Experiments were conducted using Network Simulator 2 (ns2) version 2.1b9a [19]. Theoretical simulations and non-linear least squares calculations were performed using Matlab version 5.3 Release 11 with the optimization toolbox. The Matlab lsqcurve-fit routine was called at the beginning of the simulation and as-needed to calculate gossip weights for each of the system's nodes using the static gossip algorithm in section 3.

Simulations were run to compare the Section 6.3 adaptive heuristics' performance based on cost and accuracy. Each of the test sets were adapted from the control theoretic tests in Section 6.2. Because control theory is traditionally applied to a signaling environment, the tests in that domain are based on input signals that vary over time. The tests were adapted to a network domain by varying the bit error rate occurring in the network links.

In the tests run, bit errors were injected into the system links to emulate various forms of network disruptions. Bit error rates varied between a low of 0%, corresponding to a zero signal, and a high rate of 10%, corresponding to a one signal in the control theoretic tests. The intent was to see how well the adaptive heuristics were able to adapt to varying levels of network disruption. Bit error rate settings were universally applied to all links in the network. Increased bit errors led to a rise in the number of dropped packets, which could be used to see how robust different algorithms were in the face of such disruptions.

The 10% bit-error rate is higher than expected in most wired networks, but is not unreasonable in many wireless networks, particularly when mobile. Also, if a protocol performs well with a 10% bit-error rate then it indicates protocol robustness in other challenged environments.

The tests in Section 6.4 compare the performance of the AGG protocol using the Section 5's adaptive heuristics against TCP and UDP-based implementations. In the TCP implementation, the sender (publisher) reliably sends all new information in the system to each of the group members via TCP. The UDP implementation sends all new information to the group's nodes unreliably using UDP.

Tests were performed using four 20-member subgroups linked together in a mesh, as shown in Fig. 5. The sender (the diagram's Data Source) periodically created new messages at a rate of 20 messages per round. The target data rates of 25%, 50%, 75%, and 100% were used in the tests described in Section 5.3. The same basic configuration was used in the tests in Section 5.4 except that all target rates were set to 100%. Tests started at time 3.5 and ended at time 364.0. The initial 3.5 and final 2 seconds of simulation time served as ramp-up/ramp-down time where no new messages entered the system. The runtime, ramp-up, and ramp-down periods were chosen to be large enough to demonstrate the stability, or instability, of the protocols tested. The buffers in each node, used to keep the node's known information updates, had a size of 400 messages per group the node subscribed to. Messages timed out and expired after 2 seconds. There was a 100 millisecond interval between gossip rounds. All nodes set their initial gossip timers randomly within a 100 millisecond interval. 10 trials were run for each test configuration. Packet latency was set to 1 millisecond per link traversed. Each link had a capacity of 1500 Mbits/second to eliminate bandwidth as a factor in the tests conducted.

When gossiping, nodes sent packets to at least one destination chosen at random within each subgroup. If no information was available to include in a message, perhaps because the protocol had just begun, then a weights-only message would be sent. These weights-only messages were sent in order to make sure that gossip weight up-

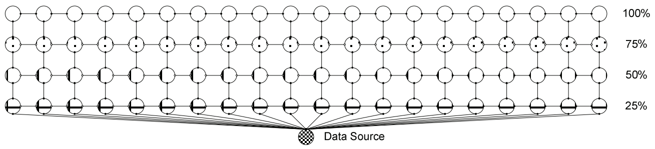

Fig. 5. The format of the adaptive tests

dates would propagate in a timely fashion.

6.2 A Description of the Control Tests and Adaptive Heuristics Evaluated

6.2.1 Network Adaptations of Control Tests

A description of the adaptive tests follows. The impulse, ramp response, and sustained oscillation tests have been adapted from control theory to the domain of communication networks. The tests are designed to measure the stability of a system. Unit step and unit step disturbance tests are not included due to space considerations. The results were consistent with those included.

Impulse: In control systems, an impulse involves applying a sudden spike from a 0 input to a 1 and then immediately cutting back to 0. In the communication tests, the network began with no bit error rate on the links. At time 123.5 all links were set to a 10% error rate. At time 243.5 the bit error rate fell back to a 0% rate. The key measurements were the time it took for the adaptive heuristics to produce gossip probability values between subgroups so that they matched their targets.

Ramp Response: This test involves injecting a ramping signal input from 0 to 1. In the experiments, the ramping begins at time 143.5 seconds. The signal rate slowly rises from a 0% to a 10% error rate at each link in a series of 20 increments. The error rate at time 143.5 + 4.0 * *t* is (0.10 / 20.0) * *t* over increments of *t* = 1 … 20. The key measurement is the system stability under shifting conditions.

Sustained Oscillation: An oscillating signal is applied to test the system's response. In the communication tests, at time 123.5 + *t* the error rate is set to $(0.10/20.0)\cdot(1.0+(t \bmod 20))$ on signal rise and $(0.10/20.0)\cdot(20.0-(1.0+(t \bmod 20)))$ on its fall. This continues for 100 steps (during which *t* ranges from 0..99).

6.2.2 Adaptive Heuristics Evaluated

The major aspects of the adaptive heuristics were described earlier in Section 5. A brief description of each of the nine heuristics is described below.

Adaptive: Uses the least-squares heuristic, described in Section 5.1, when new gossip weights are calculated.

Base No Correction: Calculates gossip weights according to the static GG algorithm. Base No Correction does not adapt gossip weights after the initial calculation.

Base Correction: Calculates gossip weights according to the static GG algorithm. Calculates the gossip weights again, using the linear least-squares heuristic, based on the difference between the static gossip weights computed and the average information rates achieved in 500 mini-trials. The mini-trials take place outside a network simulator. Every gossip message is reliably delivered to its destination and all gossip occurs all at once per round. This corrective step appears to increase the accuracy of the gossip weight calculations. Base Correction does not adapt the gossip weights after this second calculation.

Base Refinement: Calculates gossip weights according to the Base Correction heuristic. Calculates the weights again, using the linear least-squares heuristic, one time four seconds (twice the message expiration time) after gossip begins. Base Refinement does not adapt the gossip weights after this calculation.

Newton Cut Bounds: Newton's Secant method is used once the gossip weights are calculated via Base Refinement. Newton's method does not calculate new gossip weights if the observed infectivity is within +/- .02 of the target infectivity. If a calculated gossip weight is above 1 or below 0 then it is mapped to 1 or 0 respectively.

P Control: The Proportional Controller is used once the gossip weights are calculated using Base Refinement.

PI Control: The Proportional-Integral Controller is used once the gossip weights have been calculated using the Base Refinement heuristic.

The following two implementations were used as a basis for comparison against the AGG protocol using each of the previously described heuristics.

TCP: The sender establishes a TCP connection with each of the other nodes in the system. New gossip fragments are placed in each of the TCP send queues as soon as they are created at the sender. The TCP protocol is only used in the tests in Section 6.4.

UDP: New gossip fragments are sent from the sender to each of the other nodes using point to point UDP messages. UDP is only used in the Section 6.4 tests.

6.3 The Adaptive Heuristics' Performance

The adaptive heuristics in Section 5 were evaluated according to the control-based tests in Section 6.2.1. The tests consistently showed that the proportional and proportional-integral controllers outperformed the other adaptive heuristics. The heuristic based on Newton's method performed similarly to the controllers except that Newton's method had higher overall error in the scenarios run. Error here is the difference between the desired target information rate, which ranges from 0% to 100%, and the actual received information rate. Error rates are computed using a Manhattan metric, which sums the difference between the desired and actual percentage of information received for each of the four subgroups in the system. Two variations were used. In the first, the absolute values of all differences were added together and in the second, the relative differences were added.

Fig. 6 depicts key points regarding the impulse test using the absolute value metric. First, the Manhattan-metric sum of the difference between the desired and actual information rates tends to be very good in steady-state for the controllers and Newton heuristic, with a value of approximately 0.02 per subgroup. The graphs show that the non-adaptive versions of the algorithms do much worse than their adaptive counterparts. The adaptive algorithm, which uses the linear least squares method, is better than the non-adaptive alternatives, but can diverge from the target over time. Newton's method, the Proportional Controller, and the Proportional-Integral Controller perform

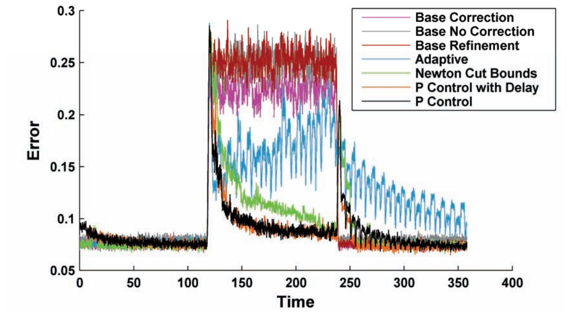

Fig. 6. Impulse Test Results based on an absolute value Manhattan metric. The graph illustrates the effect of varying bit-error rates.

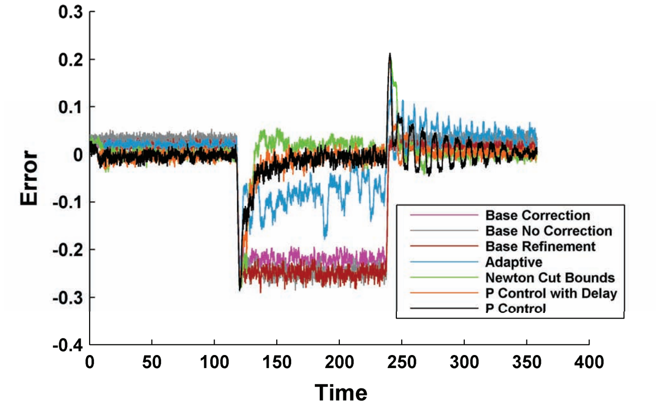

Fig. 7. The Fig. 6 impulse test, but using a relative difference Manhattan metric. The test illustrates the impact of varying bit-error rates.

best. Of these, Newton's method performs worst once the initial calibration phase ended. The Proportional Controller and Proportional-Integral Controller performed similarly. Both rapidly converge to near-zero error after a network disturbance, except that the delayed version stabilizes more quickly under shifting network conditions.

Fig. 7 shows the same impulse tests using a Manhattan metric, which sums the relative, rather than absolute, differences between the desired and actual information rates. It shows that the controllers and Newton heuristic tend to be close to the desired values and quickly stabilizes under a disruptive change from 0% to 10% bit-error rate across all links. Fig. 8 shows the result of ramp response tests using the relative Manhattan metric. As the impulse tests showed previously, the controllers and Newton heuristic are fairly stable while the least-squares heuristic oscillates and tends to remain farther away from the desired rates. The more static, heuristics do not perform well under changing conditions.

Fig. 9 shows sustained oscillation test results using the relative Manhattan metric. The graphs show the impact of oscillating bit-error rates in terms of error versus time (top) and cumulative error versus time (bottom). The results show that the controllers are relatively stable, both in terms of their rates over time and in terms of their cumulative error. The Newton heuristic also performs well. The linear least-squares (adaptive) heuristic remains within a relatively constant amount of error from the desired targets, but the heuristic also has a tendency to rise or fall on a cumulative basis over time. As expected, static heuristics perform poorly under changing conditions.

6.4 A Comparison of AGG Versus TCP and UDP-based Implementations

As mentioned in the experimental setup section, the target gossip percentage was set to 100% for all groups. One run took place with an error rate of 0% per link and another had an error rate of 10% per link. Only the cumulative error graphs for 0% and 10% fault scenarios are included in this section. The other graphs could not be included due to space limitations. In the 10% fault scenario, there were a large number of drops between nodes. Only 65% of the packets were successfully received after traveling through just four links. The key measurements were the error rate, which is expressed in terms of the difference between a perfect 100% receipt of information and the actual receipt rate, the latency between the time a new piece of information was published to the system and its receipt (computed per packet received), and the average message load in terms of the number of packets sent per second. All nine types of adaptive heuristics were used including the TCP and UDP implementations described earlier. Packets that arrived later than 2.0 seconds counted towards the total received in TCP as the protocol does not allow packets to expire.

Fig. 10 shows the difference between the desired rate of information (100% in all cases) and the actual rate. Both graphs are cumulative. The top graph shows that the adaptive heuristics, the UDP, and the TCP implementations all perform well by this metric when there are no bit-errors on the links in the system. The bottom graph depicts the cumulative error rates when a 10% bit-error rate is applied to the links. TCP's congestion mechanism prevents it from having the throughput to meet the demand for published information. The UDP-based protocol loses information, due to dropped packets, and it also has performance problems, though not as severe as with the TCP-based protocol. The AGG protocols are able to adapt to the changed conditions and still deliver nearly 100% of the published information. It is important to note that all protocols perform reasonably well in the 0% loss case. Even the worst protocol of Base Refinement only has a cumulative loss of 6 over 360 seconds. By contrast, each of the gossip variants performs far better than the roughly 50 and 250 cumulative loss rates of UDP and TCP, respectively, for the reasons outlined above. This shows the robustness of gossip in heavy loss situations.

Graphs of the average number of messages per second are not shown due to space considerations. The average number of messages sent per node remained steady in each of the protocols. One UDP packet is sent to each node in the system per published piece of information so its rate remains at a constant 20 messages per second in both the 0% bit-error rate per link and 10% bit-error rate per link cases. The AGG protocols also remain steady at a constant five messages per node per gossip round, except during adaptive gossip calibration, which take place every four seconds, when the number averages just over 6 messages per node. During the adaptive gossip calibration points, information must be gathered and disseminated regarding current system performance using gossip messages. TCP's congestion-control mechanism results in

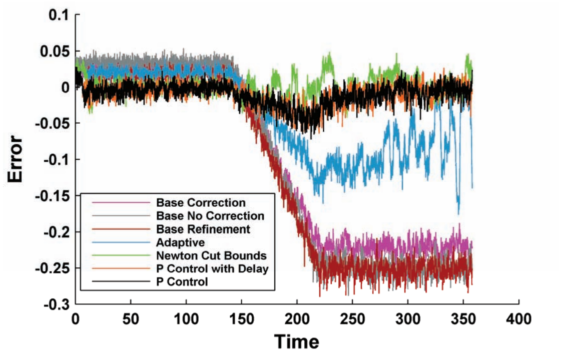

Fig. 8. This graph shows the result of ramp-response tests. The test illustrates the effect of varying bit-error rates.

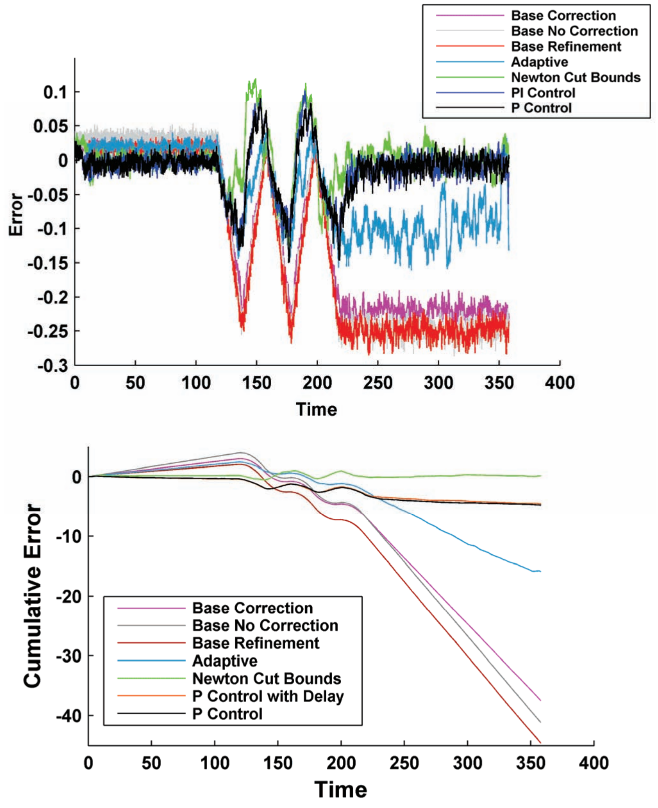

Fig. 9. These graphs show the result of sustained oscillation tests. The graphs show the result of oscillating bit-error rates. The top graph is a plot of relative Manhattan errors versus time. The bottom graph show cumulative error versus time.

a lower load on the system with an average load per node of roughly 1 packet per second rather than the 2 that is present during the 0% bit-error scenario.

Graphs depicting the latency across the 0% and 10% bit error-rate scenarios are not shown due to space constraints. The latency was extremely low for UDP packets, averaging under 10 ms, because no recovery mechanism was present. The AGG gossip-based protocols tended to have higher latency in good conditions, averaging just under 1.1 seconds. However, latency remained nearly constant as the bit-error rate increased. The TCP-based protocol, by contrast, had a latency of under 10 ms when no bit error rates were present. However, it had high latency, with an average of 28.5 seconds when bit error rates were 10% per link. Latency was the time difference between the moment an information update was first created to the time it was received via a packet at a node.

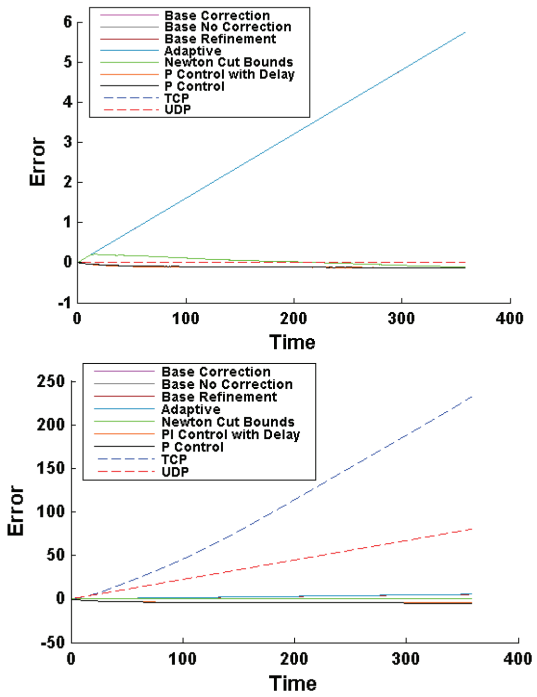

Fig. 10. The AGG, the UDP, and the TCP protocols all perform well when no bit-errors on the links (top graph). A 10% bit-error rate per link (bottom graph) is a major problem for UDP and TCP while AGG delivers close to 100% of the desired information.

These results show that AGG strikes a strong balance between delivering information at a target rate, the latency that such deliveries take, and the per node average message load. Gossip can perform better with high bit rates because of its exponential spread of information and its ability to continue to operate even when some nodes in the multicast group are receiving data at a low rate.

Previous work by the authors and others has looked at the impact of overlapping groups, of different gossip distributions, and of varying group sizes. Prior results show that system overhead decreases as subgroup sizes increase [1]. Efficiency also rises as the number of overlapping groups grows because gossip information can often be piggy-backed into combined packets [20]. Notable studies show that non-uniform distributions can improve performance when the operating environment is not uniform, perhaps due to geography or network performance differences [11, 13]. Other characteristics, such as energy use, may also be better optimized under other distributions [21]. These studies, as a whole, show that gossip can be tuned with domain knowledge for better performance and this could be a good avenue for future exploration.

7 Conclusion

This article has presented a new gossip-based group communication protocol called adaptive gravitational gossip (AGG). AGG is targeted towards publish-subscribe systems that involve periodic information updates. The protocol allows nodes to be combined into subgroups of subscribers to an information stream. Each subgroup is able to set a different target for the percent of information updates that will reach the subgroup's members. The targets are maintained with high probability despite changes in network conditions over time. The adaptive control heuristics work in conjunction with a static mathematical model of the protocol's behavior without packet losses. Applications that could benefit from the AGG protocol include Net-Centric Warfare situations in the military and in proximity-sensitive sensor networks. Experimental results illustrated the effectiveness of the control heuristics used to adapt the subgroup gossip probability values. Comparisons against basic TCP and UDP based group communication protocols illustrated high levels of expected reliability and consistent message loads in the AGG protocol. AGG shows promise for use in disruption-prone networks in situations where subgroups of nodes in publish-subscribe systems have different interest levels in a common periodic information stream. Both AGG's theoretical foundations and experimental results show that the protocol operates in a way that is fair, stable, and disruption-tolerant.

References

[1] K. Jenkins, K. M. Hopkinson, and K. P. Birman, "Reliable Group Communication with Subgroups," in *International Workshop on Applied Reliable Group Communication (WARGC)*, Mesa, AZ, USA, 2001, pp. 25-30.

[2] L. Rodrigues, S. Handurukande, J. Pereira, R. Guerraoui, and A.-M. Kermarrec, "Adaptive Gossip-Based Broadcast," in *International Conference on Dependable Systems and Networks*, San Francisco, CA, USA, 2003, pp. 47-56.

[3] B. Garbinato, F. Pedone, and R. Schmidt, "An Adaptive Algorithm for Efficient Message Diffusion in Unreliable Environments," in *International Conference on Dependable Systems and Networks*, Florence, Italy, 2004, pp. 507-516.

[4] J. V. Director for Strategic Plans and Policy, "America's Military: Preparing for Tomorrow," DoD, Ed.: U.S. Government Printing Office, June 2000.

[5] M. Gettle, "Air Force Releases New Mission Statement," in *Air Force Print News*, December 8, 2005.

[6] M. W. Wynn and T. M. Mosely, "SECAF/CSAF Letter to Airmen: Mission Statement," U.S. Air Force, December 7, 2005.

[7] U.S. Air Force Scientific Advisory Board, "Report on Information Management to Support the Warrior," Washington: HW USAF, December 17, 2000.

[8] S. Busenberg and C. Castillo-Chavez, "A General Solution to the Problem of Mixing of Sub-Populations, and its Applications to Risk and Age-Structured Epidemic Models for the Spread of AIDS," *IMA Journal of Mathematics Applied in Medicine and Biology,* vol. 8, pp. 1-29, 1991.

[9] A. J. Demers, D. H. Greene, C. Hauser, W. Irish, and J. Larson, "Epidemic Algorithms for Replicated Database Maintenance," in *Symposium on Principles of Distributed Computing*, Vancouver, British Columbia, Canada, 1987, pp. 1-12.

[10] W. E. Boyce and R. C. DiPrima, *Calculus*. New York, NY, USA: John Wiley and Sons, 1988.

[11] S.-C. J. Yam and M.-H. Wong, "Performance of Semantic-Dependent Two-Tier Gossip Mechanisms," in *IEEE Sensors Applications Symposium (SAS)*, San Diego, CA, USA, 2007, pp. 1-6.

[12] D. C. Erdil and M. J. Lewis, "Grid Resource Scheduling with Gossip Protocols," in *IEEE Conference on Peer-to-Peer Computing*, Galway, Ireland, 2007, pp. 193-200.

[13] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic Gossip: Efficient Averaging for Sensor Networks," *IEEE Transactions on Signal Processing,* vol. 56, pp. 1205-1216, March 2008.

[14] W. Jia, D. Lu, G. Wang, and L. Zhang, "Local Retransmission-Based Gossip Protocol in Mobile Ad Hoc Networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Hong Kong, China, 2007, pp. 4247-4252.

[15] The Mathworks Inc., *Optimization Toolbox User's Guide Version 3.0*, Fifth ed. Natick, Massachusetts: The Mathworks Inc., 2004.

[16] M. T. Heath, *Scientific Computing*. New York, NY: WCB/McGraw-Hill, 1997.

[17] K. Ogata, *Modern Control Engineering*, Third ed. Upper Saddle River, NJ: Prentice Hall, 1997.

[18] K. Ogata, *Discrete-Time Control Systems*. Upper Saddle River, NJ: Prentice Hall, 1994.

[19] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidermann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, "Advances in Network Simulation," *IEEE Computer,* vol. 33, pp. 59-67, May 2000.

[20] K. M. Hopkinson, "Overcoming Communication, Distributed Systems, and Simulation Challenges: A Case Study Involving the Protection and Control of the Electric Power Grid Using a Utility Intranet Based on Internet Technology," *Computer Science*. Doctor of Philosophy Ithaca, New York: Cornell University, 2004, p. 245.

[21] F. Lu, L.-T. Chia, and K. L. Tay, "NBGossip - Neighborhood Gossip with Network Coding Based Message Aggregation," in *IEEE Mobile Adhoc and Sensor Systems (MASS)*, Pisa, Italy, 2007, pp. 1-12.

Kenneth Hopkinson is an Assistant Professor of Computer Science at the Air Force Institute of Technology (AFIT). His research interests include distributed systems, networking, and simulation.

Kate Jenkins received her MS degree from Cornell University in 1999. She is currently an employee at Akamai Technologies.

Kenneth Birman is a Professor of Computer Science at Cornell University. A Fellow of the ACM, Birman has published extensively on reliable, secure distributed computing.

James Thorp is the chair of the Department of Electrical and Computer Engineering at Virginia Polytechnic Institute. He is a member of the National Academy of Engineering and a Fellow of the IEEE.

Gregory Toussaint is an Assistant Professor of Electrical Engineering at the AFIT. He is interested in control theory and its applications.

Manu Parashar is a consultant at the Electric Power Group.