

**17th ICCRTS
“Operationalizing C2 Agility”**

An Application of Information Theory to Command and Control

Topics

Concepts, Theory, and Policy
Modeling and Simulation
Networks and Networking

Authors

Kevin Schultz and David Scheidt
Applied Physics Laboratory
Johns Hopkins University
11000 Johns Hopkins Rd
Laurel, Maryland 20723
{Kevin.Schultz, David.Scheidt}@jhuapl.edu

Point of Contact

Kevin Schultz
Applied Physics Laboratory
Johns Hopkins University
11000 Johns Hopkins Rd
Laurel, Maryland 20723
443-778-3457
Kevin.Schultz@jhuapl.edu

An Application of Information Theory to Command and Control

Kevin Schultz

Applied Physics Laboratory

Johns Hopkins University

Laurel, Maryland 20723

Email: Kevin.Schultz@jhuapl.edu

David Scheidt

Applied Physics Laboratory

Johns Hopkins University

Laurel, Maryland 20723

Email: David.Scheidt@jhuapl.edu

Abstract—Recently, it was postulated that information theory would be useful in analyzing command and control (C2) processes. Here, that theory is expounded upon first by demonstrating entropic drag in sample observation scenarios and providing numerical calculations of information loss due to entropic drag. Additionally, a model for observation abstraction is introduced and an analysis between the information loss due to abstraction and entropic drag is presented. This results in optimum abstraction levels, which depend on the information-theoretic characteristics of the system being observed. Prior efforts demonstrated that the efficacy of distributed C2 system topologies vary as a function of the information-theoretic characteristics of the system being controlled. We extend that work, providing additional insight into the relationship between information theory and C2 performance. First, the parameter space is expanded, resulting in additional complexity in topology dominance. Second, we consider an alternative gossip-based communication protocol. Third, the results of an evolutionary approach to optimizing the C2 structure are discussed. Overall, these results continue the development of the application of information theory to the study of C2 and should be an important tool in the development of future C2 systems.

I. INTRODUCTION

It has been postulated that modern warfare has shifted from “platform-centric” to “network-centric,” [1]–[3], meaning the force that is able to achieve “information superiority” is better able to influence and counter the adversary. Here, information superiority relates to more *relevant* information and better situational awareness. This concept dovetails nicely with information theory [4], and the mathematics of information-theoretic control theory. Specifically, it has been demonstrated that the reduction in uncertainty achievable in controlling some system is equal to the decrease possible without using any information plus the information gathered by a controller observing the system (called open-loop and closed-loop control, respectively) [5], [6].

One difficulty in applying these concepts to command and control (C2) is that the information content (in the information theory sense) of an observation is rarely static, specifically, the reduction in uncertainty attributed to a specific observation may decrease over time as the observation becomes less

relevant to the current system state. In [7], [8] an information-theoretic and graph-theoretic characterization of a C2 process was introduced, along with a comparison of simulated information flow for a number of different C2 communication network topologies. A key concept in this characterization is the notion of *entropic information*, that is, observation outcomes whose information content monotonically decreases over time. For instance, a specific observation of a target produces information dependent on the capabilities of the sensor and encoding of the scene into coordinates (i.e., the *descriptive complexity*). As time elapses, the information content of this earlier observation decreases, and this rate of decay is defined as *entropic drag* [7], [8].

Here, the application of information theory to C2 problems is continued from [7]. Concrete examples, including numerical calculations, of systems that exhibit entropic information and are affected by entropic drag are provided. In particular, these examples provide insight into how the agility of an adversary manifests as entropic drag. Additionally, the examples illustrate how the entropic drag may be estimated, which is a necessary component in the application of these concepts to an agile C2 control system [8]. The information theory of C2 processes is also extended by looking into the information loss due to abstraction, and the trade-offs that should be considered by transmitting and processing fewer observations, when the observations are subject to entropic drag.

Another facet of the work in [7] was a demonstration of the effects of entropic drag on C2 structures, in the form of an agent-based simulation. Agents in this simulation were connected via a network that was modeled as a graph. Agents were tasked with observing some unspecified system and then passing this information to its neighbors for fusion and communication with neighbors’ neighbors, and so on. A noteworthy result of [7] was the variation in relative performance of a fixed set communication topologies as the information-theoretic characteristics of the observation problem varied.

That work is extended here by expanding the parameter space from the original set of information-theoretic parameters considered in [7]. This expanded parameter space introduces additional structure and complexity into the regions of relative dominance of the original set of topologies tested. Additionally, the same set of topologies are compared using

an alternative communications protocol, and then compared with the original broadcast model. Finally, a summary and interpretation of the results of a genetic algorithm used to optimize the topologies in the information flow simulation is presented.

II. INFORMATION THEORY AND COMPLEXITY

One theme of [7] is the notion of a cost associated with coordination and collaboration. A component of this cost is the time it takes to obtain, process, and communicate information. Not only does this increase in time tax the C2 process, this time also degrades the utility of the very information that is being manipulated. This notion of the decay of information due to the manipulation of information is called entropic drag [7], [8], and it has been shown that the information-theoretic characteristics of a scenario can determine the efficacy of particular coordination strategies [7].

To develop this theory formally, we assume that the C2 process $x(t)$ operates in a discrete state space $X = \{x_i\}$ with finite cardinality $|X|$. The discreteness of the state space can be relaxed, provided the sensor discretizes the state space in an appropriate fashion (see [8]), but for simplicity we will keep the assumption of a finite and discrete state space. The number of bits required to fully describe the state of the C2 process is the descriptive complexity, and this is equal to $\log_2 |X|$ bits. The descriptive complexity can be thought of as the fidelity at which the process is described, for instance the location of an adversary to the nearest meter as compared to kilometer.

The notions of uncertainty and information of a C2 process are quantified using Shannon information entropy [4]. The information entropy H (in bits) of the process x at time t is

$$H(x(t)) = -\sum P(x_i, t) \log_2(P(x_i, t)),$$

where $P(x_i, t)$ is the probability that the process x is in state x_i at time t . The information I of a particular observation $x(t) = x_i$ is

$$I(x_i, t) = -\log_2 P(x_i, t).$$

Thus, an interpretation of entropy is the expected information gain of an observation. It is known that entropy is maximized when all states of the system are equally probable ($P(x_i, t) = |X|^{-1}$, all $x_i \in X$). This means that the descriptive complexity is also the maximum possible entropy of the system, given a fixed discretization.

Next, consider an observation $S(x_i, t)$, interpreted as $x(t) = x_i$. Then, the initial information content of the observation (allowing for an abuse of notation) is

$$I(S, t) = -\log_2 P(x_i, t).$$

Note that the second argument in $I(S, t)$ is a time index as well. As time elapses, the relevance of the observation S (that occurred at time t) should be less at time $t' > t$. If the system is not uniquely defined by a single observation, then the conditional expected information content of a second observation $S' = (x_j, t')$ of the same sensor for time $t' > t$ is $H(x(t')|S) = H(x(t')|x(t) = x_i)$ and this is non-zero. The

very fact that repeating the same observation (i.e., polling the same sensor) results in the gain of new information beyond the original observation indicates that the previous observation's information content has in some sense decayed. For a sequence of k observations $S_{1:k} = \{(x_j, t_j)\}$ of the process x ending at $t_k = t$, we define the entropic drag (Γ) of the system on the observations $S_{1:k}$ at time $t' > t$ by

$$\Gamma(S_{1:k}, t, t') = \frac{H(x(t')|S_{1:k})}{t' - t}.$$

Conceptually, entropic drag can be thought of as the time derivative of conditional entropy, but in a strict mathematical sense the assumed discrete space will not admit a derivative. For observations that occur with fixed sampling time $\Delta t > 0$, the quantity $\Gamma(S_{1:k}, t, t + \Delta t)$ is effectively the expected rate of information generation of the system at time t . This notion can be generalized to multiple sensors, but requires a more complicated exposition and the single sensor case is sufficient to illustrate the concepts here (see [8] for this exposition). Entropic drag should not be interpreted solely as the change of state of the underlying system. For example, a pendulum or train moving at a fixed speed have predictable trajectories and have considerably lower entropic drag that systems whose motion is not as constrained.

A. Entropic Drag for Target Tracking

The definitions of entropic drag and related concepts in [7], [8] appeal to basic intuition about the behavior of real-world systems, but the entropic drag of specific example systems was not calculated. Here, we provide calculations demonstrating entropic drag using an example C2 system, that being a target tracking scenario. In this tracking scenario, a single target is tracked in discrete time in a 51×51 zone "world", and the sensors are noiseless and observe the entire state space at once. Thus, $x(t)$ is the location of the target, and its domain X are the 51×51 discrete zones that it can inhabit. Furthermore, the sampling rate and target motion occurs at discrete instances, and to emphasize this, we write $x(t) = x(k)$, for $k \in \mathbb{Z}_{\geq 0}$.

A number of different probabilistic motion models are considered. The motion model of the target corresponds to the probabilities assigned to the location of a target at time step $k + 1$, given its location at time k . Thus the motion model is Markovian. To find the probability distribution of a target at a later time, given its distribution at time k , we repeatedly apply this motion model at each zone, and sum these weighted by the probability that the target was in that zone at the previous time step, with circular boundary conditions assumed.

Formally, for any zone x_i in the state space X , let $P_{x_j}(x_i)$ be the probability that the target moves from the zone x_j to the zone $x_i \in X$ in one time step. As noted above, the targets position at the next time step is dependent only on its current position. Next, let $P(x_i, k)$ denote the probability that the target is in zone x_i at time k . Then, we have that

$$P(x_i, k + 1) = \sum_{x_j \in X} P(x_j, k) P_{x_j}(x_i).$$

Given some prior $P(x_i, 0)$ (for these results, a uniform prior of $P(x_i, 0) = 1/|X|$ was used), the probabilities of target state can be propagated forward in times by repeatedly applying the update rule.

The uniform prior provides no initial information about the location of the target, so the uncertainty before any observations is $\log_2(51 \cdot 51) = \log_2(2601) \approx 11.34$ bits. The sensor is assumed noiseless, so the initial information content of the first observation is ≈ 11.34 bits, regardless of the motion model for the target. This initial observation is the only observation performed, and the effects of entropic drag on the information content of this single observation are calculated. However, the different motion models cause the information content of this initial observation to decrease at different rates. Again, since sensor is noiseless and can observe the entire state space, the observation x^* returned by the sensor creates a probability field $P(x, 1)$ where

$$P(x, 1) = \begin{cases} 1 & \text{if } x = x^*, \\ 0 & \text{else.} \end{cases}$$

To calculate the information decay from this single observation, the motion models (that is, the state transition probabilities) need to be specified. The models considered here are:

- 1-zone movement: target moves in a known direction.
- 5-zone movement: target moves up, down, left, right, or stays put with equal probability.
- 5-zone drift: target moves up, down, left, right, or stays put with unequal probabilities.
- 9-zone movement: target moves to an adjacent zone (including diagonals) or stays put with equal probability.
- 9-zone drift: target moves to an adjacent zone (including diagonals) or stays put with unequal probabilities.
- 25-zone movement: target moves up to two zones in any direction (including diagonals) or stays put, all with equal probability.

Figure 1 shows the information content of the single observation over time for each of the motion models. Since entropic drag is roughly equivalent to the rate at which the information content decreases, this plot actually shows the integral of this decay rate.

The one-zone movement model is a non-entropic system. Initially, the target's location is unknown, but once an observation is performed, its location is perfectly predictable (as the direction of motion is assumed to be known by the observer). Thus the information content of the observation does not decay over time. The difference between the five-zone movement and five-zone drift models is the probability distribution on each of those five options. In the five-zone movement model, the probabilities are all uniform, so this motion model introduces maximal uncertainty at each iteration. Note that the five zones need not be the specific five zones, only that $P_y(x)$ is uniformly distributed among five zones for each y . The five-zone drift model has an unequal probability distribution on the same five zones, and thus at

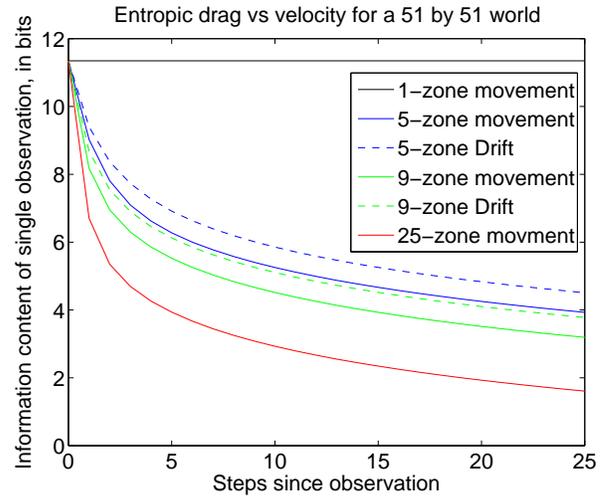


Fig. 1: Plots of information decay as the observation rate decreases relative to the movement rate of the prey. Alternatively, plots show the remaining information content of a prior observation.

each iteration less uncertainty is added (i.e., less information is lost from the initial observation) as compared to the five-zone movement model. The nine-zone movement model is a uniform distribution on nine possible zones, and the nine-zone drift model is skewed to introduce less entropy at each iteration. The nine-zone drift model was chosen to introduce more uncertainty than the five-zone movement model, but this is not necessarily required (i.e., there are distributions on nine zones that introduce less uncertainty at each iteration than the uniform distribution on five zones). The 25-zone movement model, introduces the most uncertainty at each iteration among the motion models considered.

Since all of these models apply the same movement rules at each iteration, the state transition probabilities are spatially invariant. This causes the relative orderings of the information contents associated with a single observation of each movement model to stay fixed. This is not always the case, as the state transition probabilities model could have dependency on the current zone state of the system. This could have a drastic effect on the information decay calculations. For example, consider a target that only occupies a few zones, but moves randomly between them. Additional variation in the decay could result if the systems next state was dependent not only on the current states, but also on prior states (i.e., the system is not memoryless), or if the transition probabilities are dependent on time (i.e., the system is non-stationary).

There is another class of movement models that is not considered in Figure 1, those that are non-entropic when conditioned on multiple observations. Consider the following target movement model for the same scenario as in Figure 1: the target moves in the same direction at each iteration but this direction is initially unknown (cf., the one-zone movement model). In this situation, any single observation considered by itself will decay based on constant motion in each of

the nine directions uniformly. So after the first observation, the targets location will be in one of nine known squares at each subsequent iteration. So the initial observation loses information content, but only after the first iteration. A subsequent observation that didn't also consider the previous observation would gain this information back, only to lose it at the next iteration when the target moves. However, if two observations are conditioned upon, then the target motion is known and the information is non-entropic. It is easy to see how more complex target movement models requiring multiple observations to make the information non-entropic could be constructed.

It should be noted that Figure 1 is not the result of some Monte Carlo simulation, but is a numerical calculation of the target probability field using the motion model. For illustrative purposes, the targets all started in the exact middle of the tracking space, and we truncated the calculation well before any boundary effects occurred. The calculation of probability of target presence is not particularly difficult, provided a mostly spatially identical motion model. In fact, it is a convolution between the current probability field and a kernel corresponding to the motion model. Corrections must be performed, however, for boundary conditions. In particular, truncated, toroidal, and reflective boundary conditions all undergo different correction calculations.

The information decay shown in Figure 1 can be interpreted in two ways. The first, as noted above, is that it can be viewed as the information loss due to time elapsing. This delay could be introduced by transmission delay in the communication, delay introduced while the sensor's operator is performing other tasks, or any number of reasons. The second interpretation is the effects of entropic drag induced by increased target mobility, i.e., if the movement rate of target is greater than the sampling rate of the sensor, then the information loss between observations is greater.

This second interpretation highlights the impact of agility on the observer, orient, detect, and act (OODA) loop [9]. If the target is more agile, then the information content of observations decreases faster, and the associated processing of observations associated with that target could result in less useful actions taken. Furthermore, if the target were performing its own OODA loop targeting the tracker/sensor, then the extra time spent by the tracker to gather information about the target could allow the target get inside the OODA loop of the tracker and successfully execute a counterattack on it.

While this tracking scenario is rather simple, these concepts are apparent in a number of research areas related to state estimation. The Kalman filter is a classic technique for tracking, data fusion, and more generally recursive state estimation [10]. Uncertainty in the Kalman filter is represented by the propagated covariance matrix. The difference (in some sense) between successive covariance matrices relate to the information gained by each observation, assuming observations are taken at each time step. When observations are not taken at each time step (relative to the prediction step of the filter),

this is known as intermittent Kalman filtering. There are a number of results that speak to the change of uncertainty over time relative to different rates of sampling [11]–[13]. This idea can be extended to multi-target/multi sensor scenarios, for example, through the notion of random finite set based tracking [14], [15]. These latter concepts apply even when the number of targets is also uncertain and time-varying.

Given the current distribution of state for the C2 process and its transition function, the calculation of entropic drag is in theory straight forward. However, state estimation and functions of estimated state are generally not as easy. Since the observation space considered here is discrete, quantized particle filtering or discrete space hidden Markov models may be useful in the development of estimators of entropic drag [16], [17]. Robust and efficient estimation of the information-theoretic properties of the system being observed is an important part of an agile C2 system. Spending time to communicate and process irrelevant information further degrades existing information. The ability to dynamically decide with whom to communicate and accurately compute the utility of potential information is a key factor in maximizing useful information and thus situational awareness.

B. Abstraction in Complex Scenarios

In [18] the effects of increasing the descriptive complexity of an underwater C2 scenario were investigated. The descriptive complexity was increased by increasing the spatial fidelity of the world. Using a communications channel with fixed bandwidth and a fixed geometric rate of information decay (entropic drag), it was illustrated that there is an optimal level of spatial fidelity after which the increased communication time results in less overall information, due to the effects of entropic drag.

The results in [18] illustrate a trade-off that needs to be considered when entropic drag occurs, namely the balance between the loss of information incurred due to a reduction in descriptive complexity as compared to delay introduced by dealing with the scenario at high fidelity. Alternatively, the spatial fidelity of the scenario could be held constant, and the elements of interest in the scenario abstracted. By abstraction, we mean the process of summarizing a number of elements of interest into a broader element. This is related to fidelity of another sort, namely, the aggregation of high-resolution states into a single state that conveys similar (but potentially less) information. This generalization into a single macroscopic state carries with it an additional amount of uncertainty. For example, individual soldiers' positions could be grouped into an approximate squad position, squads into platoons, platoons into companies, and so on. First of all, observations may be abstracted to a certain level (say the company level), because the additional fidelity of a more fine level of detail does not present much additional utility to the operators of some C2 process. Alternatively, perhaps the information sources themselves are not capable of producing finer detail. Here, however, we do not consider the reason

for abstraction and instead focus on the interaction between abstraction and entropic drag.

Abstraction may result in a loss of information, but it also results in less observations to manipulate. As was shown in [7], [19], the time-cost of communicating and processing a set of observations reduces the information content through entropic drag. Thus, there is a trade-off between abstraction and entropic drag, resulting in an optimal level of abstraction in a given scenario. This optimal level depends on the rate at which information content decays, so the information-theoretic characteristics of the system are important in determining this level of abstraction.

Consider the following model to illustrate this phenomenon. There are $N = 128$ elements of interest, and the world is set up such that elements are described using messages of length $L = 64$ bits. Thus, the maximum descriptive complexity in this scenario is $N \cdot L = 8192$ bits. However, there is the option of describing elements using less than N messages (abstracting them), but a loss of information is incurred in doing so. To calculate the cost of abstracting to $M \leq N$ messages, assume that the information content of each message decays proportionally to the square of the number of elements per message, N/M . This assumption is motivated by tracking in a two-dimensional environment, if the 64-bit message length is directly related to the resolution of the sensor (i.e., the world is divided into 2^{64} distinct locations), then we expect that n elements would be distributed spatially proportionally to n^2 . In this scheme, the information content of M messages is

$$I(M) = N \cdot L - \lfloor N/M \rfloor \cdot \log_2(M^2/N^2).$$

Figure 2 shows calculation of $I(M)$ for M taking values between 1 and 128. For these assumptions, $I(1)$ still retains a significant portion of the possible information, but this is adjustable by changing the expression $\lfloor N/M \rfloor \cdot \log_2(M^2/N^2)$ to one that penalizes low message counts more harshly. Also, as the number of messages used increases, there is less information gain per additional message, indicating diminishing returns. On its surface this model may seem to violate some tenets of information theory by claiming that it is possible to summarize many elements through abstraction and create a single message that is more informative than a message about a single element. However, this model is exploiting the correlations (i.e., mutual information) between the elements to create messages that are more informative (i.e., changing the coding of the messages).

Note that $I(M)$ only accounts for information loss due to describing N elements in M messages. If these M messages are then communicated, entropic drag from the resulting time lapse for the communication of M messages further reduces the information content of the original observations. We will assume that communication occurs over a noiseless channel, and there is an ideal coding that allows for the messages to be sent using L bits per message. Let this channel have bit-rate r bits/sec, then it takes $T = L/r$ seconds to transmit a message. To quantify entropic drag, we use a geometric decay rate of γ . Thus, the final value of the information received

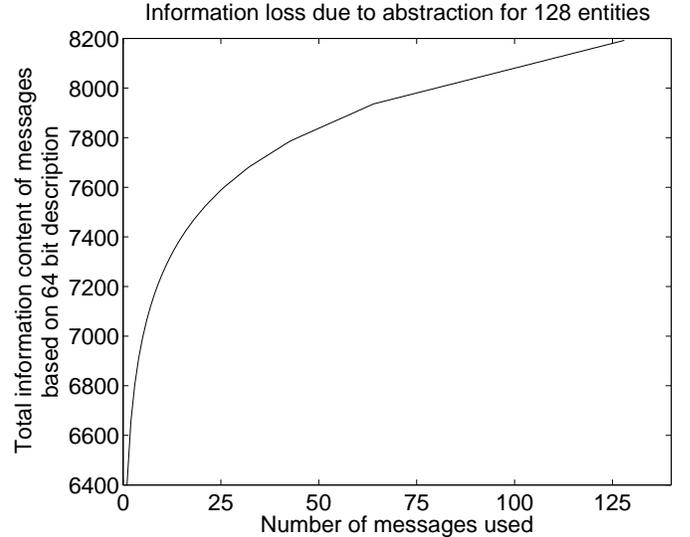


Fig. 2: Abstraction model information loss due to abstraction.

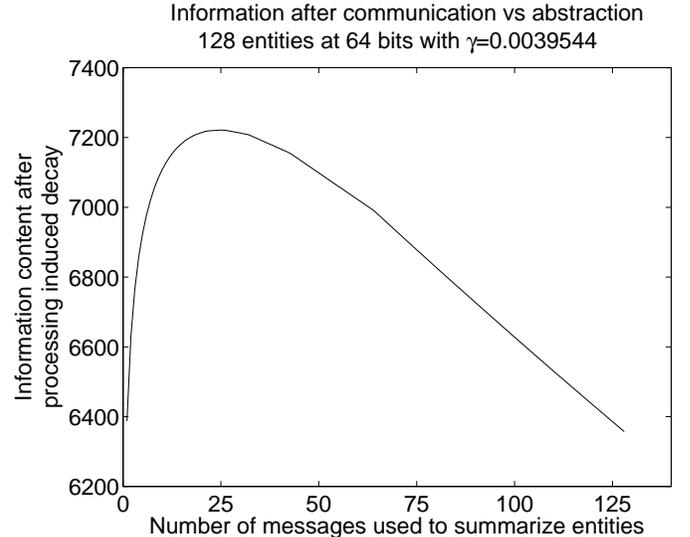


Fig. 3: Information loss due to abstraction and entropic drag during communication of messages, for a specific geometric decay approximation to entropic drag.

after both abstraction and communication is

$$\tilde{I}(M, \gamma) = I(M) \cdot (1 - \gamma)^{MT}.$$

The value of γ has an interesting affect on which level of abstraction is optimal. As $\gamma \rightarrow 0$, \tilde{I} becomes a monotonically increasing function of M , indicating that for low enough γ , $M = N$ is optimal. As $\gamma \rightarrow 1$, this function will become monotonically decreasing, and a single message is optimal. This transition is smooth, so for a range of decay values, there is an optimum M between 1 and N . An example of a decay rate that results in an optimum level of abstraction is shown in Figure 3, for communications bandwidth $r = L$ bits per second and geometric decay rate $\gamma \approx 3.95 \times 10^{-3}$.

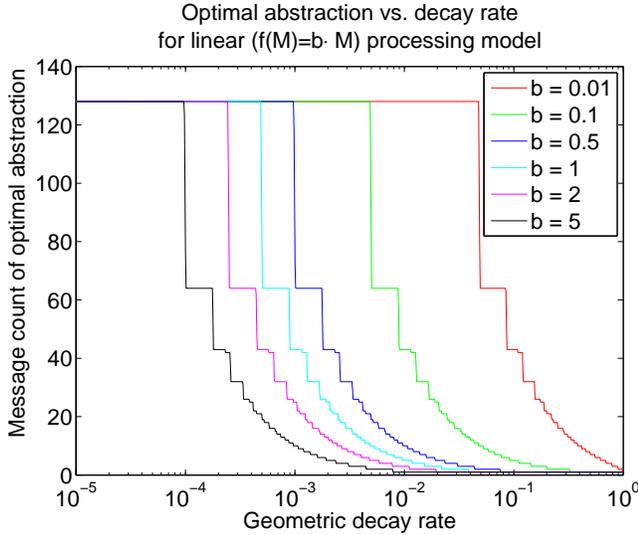


Fig. 4: Optimal abstraction level at different geometric decay rates for various linear processing time costs. Note the logarithmic scale of the decay rate.

This notion of abstraction is not only applicable to communication, but also to the processing and manipulation of information locally. For a given M , suppose we have a function f that indicates the amount of processing time that it takes to deal with M elements. Then, the ‘ MT ’ in \tilde{I} can be replaced with $f(M)$. For example, consider a linear processing model $f(M) = bM$. This is the same model used in [7], and represents a relatively low-level of computational complexity. Alternatively, a linear processing model could represent the time it takes to communicate messages for the M elements, transmitted serially. The effects of varying the slope b in a linear processing model on the optimal abstraction level (in terms of the number of messages used in the abstraction) are shown in Figure 4. As expected, larger values of b favor higher levels of abstraction (fewer messages) for a given rate of information decay. What is somewhat surprising, however, is the sharp drop between no abstraction initially, to summarizing the scenario using half of the maximum number of messages. No levels of abstraction in between these two are optimal over the decay rates and slopes investigated. Similar, but smaller, drop-offs occur as the decay rate increases for each value of b . These results indicate that the optimal abstraction level is sensitive to both the processing model and rate of information decay. Figure 4 only shows the optimum abstraction level, Figure 5 shows the information content for the same linear processing models at the optimum level of abstraction.

The optimal levels of abstraction at a given decay rate for additional processing models are shown in Figure 6. These processing models (in order of increasing complexity) are linear, log-linear, quadratic, quartic, exponential, and factorial complexities. These increases in complexity result in even more abstraction than slope increases in linear complexity.

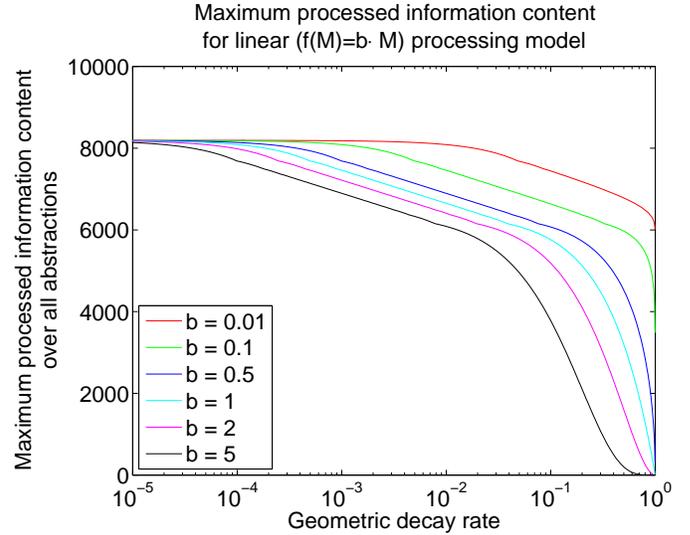


Fig. 5: Maximum information content after decay associated with processing time for various linear models. Note the logarithmic scale of the decay rate.

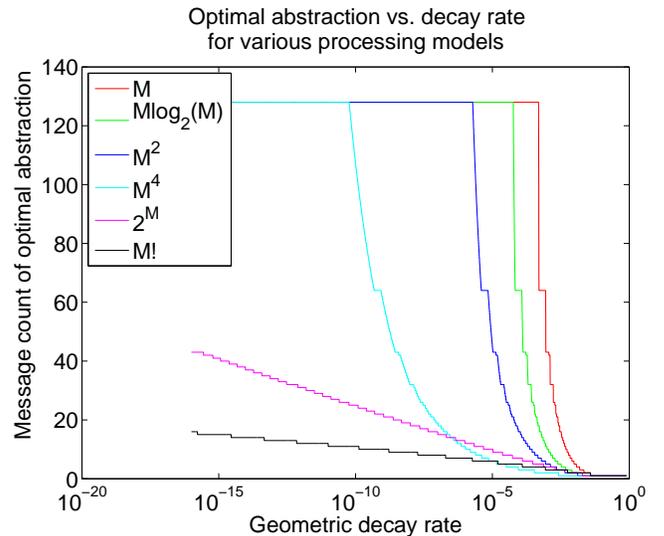


Fig. 6: Optimal abstraction level at different geometric decay rates for a number of different processing models. Note the difference in x -axis as compared to 4 and 5.

III. SIMULATION-BASED EXPERIMENTS OF DISTRIBUTED SENSOR NETWORKS

The previous section continued the development of the application of information theory to the study of C2. The remaining portions of the paper present several results of simulation-based experiments derived from a simulation of [7].

A. Problem Definition

As in [7], the simulation considered here models a scenario of multiple agents observing a common system. The agents are trying to maximize the amount of information about the system that is distributed amongst the agents. Some of the

agents are equipped with sensors, and these sensors observe distinct portions of the system, so there is no mutual information between sensors. The agents' communication network is modeled as a graph. As in [7], the motivating question is which network topologies perform well in which regions of information-theoretic parameter space of the system being observed.

B. Graph Theory Preliminaries

As the communication network of the agents in the simulation is modeled using a graph, some graph theory preliminaries are introduced that cover the concepts required here. Let a graph \mathcal{G} be defined by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ are the vertices of the graph and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the edges of the graph. We assume that \mathcal{G} is undirected and there are no self-loops, so $(i, j) \in \mathcal{E} \iff (j, i) \in \mathcal{E} \forall i, j \in \mathcal{V}$ and $(i, i) \notin \mathcal{E}, \forall i \in \mathcal{V}$. Let $\deg(i) = |\{(i, j) \in \mathcal{E} : j \in \mathcal{V}\}|$ be the degree of the vertex i . A graph is said to be connected if for any $i, j \in \mathcal{V}, i \neq j$, there is a sequence of vertices $a_k, k = 0, \dots, K$, where $(a_{\ell-1}, a_\ell) \in \mathcal{E}$ for $\ell = 1, \dots, K, a_0 = i$, and $a_K = j$. We call this set of vertices a path, and the length of the path is said to be K , and the minimum length of all such paths between i and j is called the (geodesic) distance between i and j . Denote this length $d(i, j)$ when i, j are connected, and define $d(i, i) = 0$, and $d(i, j) = \infty$ if they are not connected. A path is said to be simple if it repeats no vertices from start to end. A graph is said to be acyclic if for every $i \in \mathcal{V}$, there does not exist a sequence of vertices $a_k, k = 0, \dots, K$ such that $(a_{\ell-1}, a_\ell) \in \mathcal{E}$ for $\ell = 1, \dots, K, a_0 = a_K = i$, and $a_1 \neq a_2 \neq \dots \neq a_{K-1}$, i.e., there does not exist a simple path of non-zero length from any vertex i to i .

A fully connected graph is a graph where each vertex is connected to all other vertices, i.e., $\deg(i) = |\mathcal{V}| - 1, \forall i \in \mathcal{V}$. A tree is a connected acyclic graph. If one vertex i in is designated the "root" of the tree, it is a rooted tree. The parent of a vertex i in a rooted tree is the vertex j such that $(i, j) \in \mathcal{E}$ and (i, j) is in the simple path to the root, and every vertex except the root has a unique parent. The children of a vertex i are the set of vertices for which i is a parent, and a leaf is a vertex without any children. Vertices of the same geodesic distance from the root are said to be of the same generation. Two classes of rooted tree are included in [7], m -ary trees, where each vertex has at most m children, and regular trees that are described by a vector $[a_1, a_2, \dots, a_n] \in \mathbb{N}^n$, where the root has at most a_1 children, its children have at most a_2 children, and so on. Clearly an m -ary tree is also an $[m, \dots, m]$ regular tree.

A path graph is a tree with two vertices of degree one, and the remaining vertices of degree two. A 1-ring is formed by taking a path graph and adding an edge between the two vertices of degree one (assuming $|\mathcal{V}| \geq 3$). A k -ring (for $k > 1$) can then be defined by connecting each vertex to vertices with geodesic distance $\leq k$ along the 1-ring, up to $k = \lfloor |\mathcal{V}|/2 \rfloor$, where $\lfloor x \rfloor$ is the greatest integer $\leq x$. The final class of non-random graphs considered in [7] were two-dimensional grid graphs. A grid graph \mathcal{G} is defined as the

Cartesian product of two path graphs $\mathcal{P}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{P}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, where the vertices of $\mathcal{G} = \mathcal{V}_1 \times \mathcal{V}_2$ and two vertices (i, i') (j, j') are adjacent in \mathcal{G} if $i = i'$ and $(j, j') \in \mathcal{E}_2$ or $j = j'$ and $(i, i') \in \mathcal{E}_1$.

Two different classes of random graph were included in [7]. Small-world graphs were introduced in [20], but the variation in [21] was used in [7]. In this variation, a small-world graph is generated from a low-dimensional lattice by adding additional edges to the graph, instead of "swapping" a vertex in an existing edge. Using k -rings as defined above for the base lattice, denote the resulting small-world graph as a k -ring with m additional links (k -ring+ m for short). The other class of random graph is the scale-free graph generated using the preferential attachment mechanism of [22]. Let (ℓ, m) scale-free denote a graph generated starting from a fully connected graph of ℓ vertices and adding ℓ vertices each with degree m for each iteration of the construction process.

C. Simulation Review

In this simulation, N agents are represented as vertices in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $|\mathcal{V}| = N$. There is a set $\mathcal{S} \subseteq \mathcal{V}$ of the vertices called sensors and these sensors are modeled as exogenous communications channels that are observing some system of interest, and let $N_S = |\mathcal{S}|$.

These agents are then implemented using a discrete event system (DES) model [23] that can accommodate the variable delays in computation and communication that form one element of investigation here. In this simulation, each agent is in one of four states: SEND, CHECK, COMPUTE, or SENSE, representing the agent sending, checking, processing, and sensing new information, respectively. In the sensing state, an agent $i \in \mathcal{S}$ reads one bit of information and associates with this information a time-stamp at the current simulation time. In our experiments we assign an arbitrary delay of one second to the sensing state. From the sensing state, an agent next enters into a computation state whose time length is a function of the number of observations to be processed. Computational complexity is defined by a linear delay that is proportional (β sec per observation) to the number of new sensor readings, counting each sensor no more than once (i.e., only the most recent observation from a given sensor is "processed."). After COMPUTE has been completed, the agent enters SEND to communicate to all its neighbors in the graph \mathcal{G} by sending observations that have been received and/or sensed (as applicable). This takes the same amount of time as the SENSE state. From SEND the agent enters SENSE again if the agent is in \mathcal{S} and there are still sensor readings to perform, otherwise, it goes into CHECK. CHECK is essentially a holding state where the agent remains until it receives new observations, when the process transitions to COMPUTE on the new observations. The CHECK state checks for new observations to process every 1 sec. The state transitions are shown in Figure 7. This simulation is inspired by the OODA loop popular in military strategy [9], and in particular encompasses the first three steps of the loop.

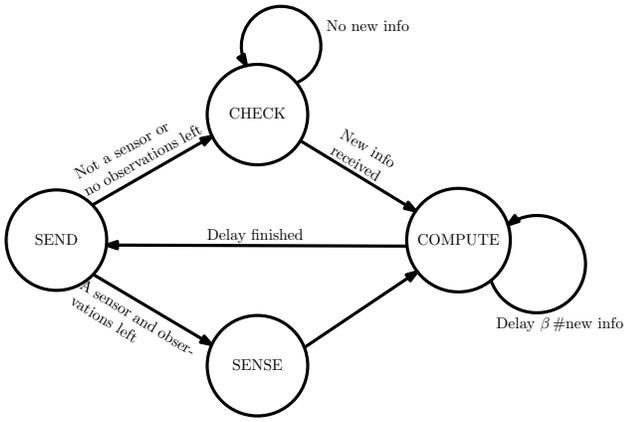


Fig. 7: State diagram for an agent in the DES simulation, using a linear complexity factor β .

For the simulations considered here, sensors observe one bit each of independent information simultaneously at the start of the simulation, and the remaining non-sensor agents are in the CHECK state. Each sensor makes only this single observation. This experimental setup was chosen to investigate a topology's response to a single anomalous event. As noted above, the information content of an observation is initially one bit, but entropic drag is modeled by applying a geometric decay rate Γ (in bits/sec) to the information content of an observation. Thus, for some time t after an observation, the information content of that observation is $(1 - \Gamma)^t$. Given that the information content of an observation decays over time, define the *information volume* at a point in time to be the sum over all agents of all the information content of observations received up to that point in time. Clearly, in this setup, the information volume is initially zero and bound by $N \cdot N_S$. Next, define the *processed information volume* at a given point in time to be the sum over all agents of information content whose associated observations have gone through that agent's COMPUTE stage by that time. The principle metric of interest in this simulation is the *peak processed information volume*, that being the maximum over time of the processed information volume.

As was the case for the first simulation in [7], all topologies considered here have $N = 127$ vertices and $N_S = 64$, including the outputs of the optimization algorithm (see Section III-F). These parameters correspond to the size of a full binary tree with seven generations having all of the leaves as sensors. All of the same graphs with identical sensor placements that were studied in [7] are considered here. This includes a fully connected graph; a star graph ([126] regular tree); the aforementioned binary tree ($[2, 2, 2, 2, 2, 2]$ regular tree) truncated $[3, 3, 3, 3, 3, 3]$, $[4, 4, 4, 4]$, $[2, 3, 4, 5]$, and $[11, 11]$ regular trees; 1-, 2-, and 3-rings; two each of (1,1), (3,3), and (5,5) scale-free graphs; two each of 1-ring+5, 1-ring+10, 3-ring+5, 3-ring+10, and 3-ring+15 small-world graphs; three 1-ring+15 small-world graphs; and a truncated 11×12 grid graph. For completeness, also considered here is the completely disconnected or isolated graph.

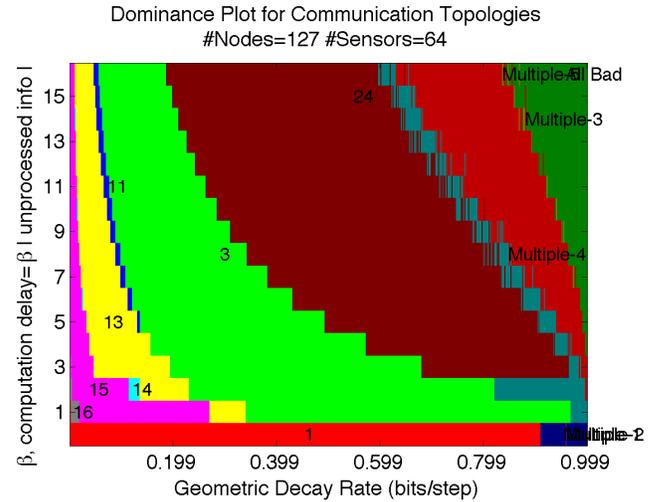


Fig. 8: (color online). Regions of dominance for different topologies in an information flow simulation. 1) fully connected, 3) binary tree, 11) (1,1) scale-free 13) (3,3) scale-free 14) (3,3) scale-free 15) (5,5) scale-free, 16) (5,5) scale-free, 24) 3 ring+10, Multiple 1&2) various regular trees, Multiple 3) 3-ring+10, 3-ring+15, grid Multiple 4) 3-ring+10, 3-ring+15, Multiple 5) several graphs, All Bad) all graphs equally poor. Reprinted from [7] with permission from the author.

In [7], 31 different topologies were compared. As the processing requirements and the rate that the information became irrelevant (see Section II) were varied, the relative performance of different topologies varied. A figure showing how the best performing topology varied as the two information-theoretic characteristics varied is shown in Figure 8. In general, when the processing requirements β and decay rate Γ were low, scale-free graphs [22] performed the best out of the different topologies examined. As the processing requirements took more time or as the decay rate increased (or both), graphs with lower degree centrality [24] began to perform better, first with the binary tree, then small-world graphs based on ring lattices [21]. Finally, there was a region where all of the graphs performed equally poor.

D. Topology Performance for Non-Integer Delay

The results of [7] shown in Figure 8 were generated using only integer values for the complexity factor β . When non-integer values of β are considered (incremented between 0 and 16 in 0.25 intervals), additional structure is apparent as shown in Figure 9. The notion of non-integer delay simply means that the processing time for an observation is not an integer multiple of the sampling time. In [7], the sampling time (SENSE) was the same as the communication time (SEND) and the hold state (CHECK) polling time (as is also the case here), and when the computation delay factor β is an integer multiple of the sampling time, the time for the COMPUTE state is also an integer. In this case, the system is effectively a discrete time system, and the sampling time is the time step. When β is not an integer, the system becomes in some sense

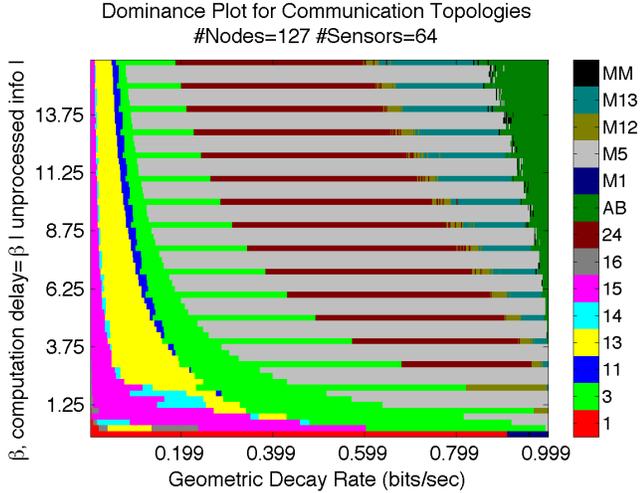


Fig. 9: (color online). Plots of topology when non-integer values for β are considered. Numbered dominant topologies 1-24 are the same as in Figure 8, M1) various trees, M5) both (1,1) scale-free graphs, M12) 3-ring+10 and 3-ring+15, M13) 3-ring+10 and 3-ring+15 and grid, MM) A combinations of twelve different groups of graphs whose total dominance was less than 0.2% of the overall decay-complexity values tested, AB) all topologies performed equally poorly.

more asynchronous as the agents now potentially have state transitions at non-integer times, once they have completed a COMPUTE state.

The banding effect of Figure 9 is such that the topologies dominant at integral values in the bands are only dominant at integral β values, even when very small increments (0.001) in complexity are considered. This appears to be related to the fact that in the integral computational delay cases, all of the states will have integer time delays. It appears that in the non-integer computational delay cases, the highly non-uniform nature of the (1,1) scale-free graphs are able to exploit the less synchronized state transition times to outperform the more uniform small-world and binary tree topologies for some ranges of β and Γ . This increase in relative performance in the regions of non-integer delay is not particularly large, as the (1,1) scale-free graphs are the second best performing graphs in the striated region, even when the delay factor is an integer.

E. Efficiency of Gossip Protocols

In the original formulation of the simulation in [7], an agent in its communication stage “pushed” observations to all of their neighbors. An additional communication protocol is considered here that is gossip-based [25]. In this gossip-based protocol an agent chooses uniformly at random one of its neighbors to communicate with, as opposed to the original simulation, where all neighbors in the network were communicated with simultaneously.

Formally, if the communication topology is the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ then when the agent i enters its SEND stage in the original simulation, it sends all of agent i ’s most up-to-date

observations of each sensor (both those observed directly in the SENSE state and those received in communication with other agents) to each agent j such that $(i, j) \in \mathcal{E}$. All of this communication occurs simultaneously in the SEND state. In the gossip protocol, an agent i in the SEND state selects a single agent j to communicate with. This agent j is such that $(i, j) \in \mathcal{E}$ (i.e., j is a neighbor of i), and the probability of selecting j is such that $P(j) = |\{j : (i, j) \in \mathcal{E}\}|^{-1}$. Again, all of agent i ’s most recent sensor observations are communicated to j . Note this is time-independent, so the agent i does not take into consideration which of its neighbors that it has previously communicated with.

This gossip protocol potentially reduces the amount of communication in the network initially (and would continue to do so should repeated observations be taken), and in particular reduces the number unprocessed observations that each agent must process in a given computation cycle. Since agents are processing fewer observations per COMPUTE stage, this stage takes less time. When entropic drag is high, this could potentially result in more processed information volume, as the processing of more information would decay to the extent that the additional observation’s increase in information could not offset the additional decay. However, when entropic drag is low, the inefficiency of the gossip-protocol as compared to the original communication method should result in poorer performance.

This communication protocol was implemented in a simulation similar to [7] but the SEND state was changed to implement the gossip protocol described above. Other than this change, the simulation is otherwise comparable to the earlier simulation in [7]. As in [7] the simulations here have single observation taken by the sensors at the start, and the task is to see how this initial set of observations progresses through the agents in the simulation. Also, the range of decays (0.001 to 0.999 in 0.002 increments) and complexities (0 to 16 in 0.25 increments) used in Section III-D (see Figure 9) are used here. Unlike the earlier results of [7] which were deterministic (except for the creation of small-world and scale-free graphs, which were first sampled, and then effective samples held constant throughout the series of experiments), the gossip-based experiments are inherently random. To account for this, 100 Monte Carlo simulations were performed for each information decay and processing delay pair for each of the 31 original topologies tested in [7].

For the deterministic simulations of [7], the metric of comparison between topologies is the peak processed information volume (see Section III-C). For the gossip-based simulations, however, the mean peak processed information volume over the Monte Carlo simulations is the metric of comparison. This quantity is used to compare topologies using the gossip protocol the peak processed information volume of the deterministic simulations (and with the mean peak processed information volume of other gossip-based communication simulations). Due to the increased number of dominant topologies and co-dominant topologies, as well as the fragmentation of dominant regions due to the randomness of the gossip communication

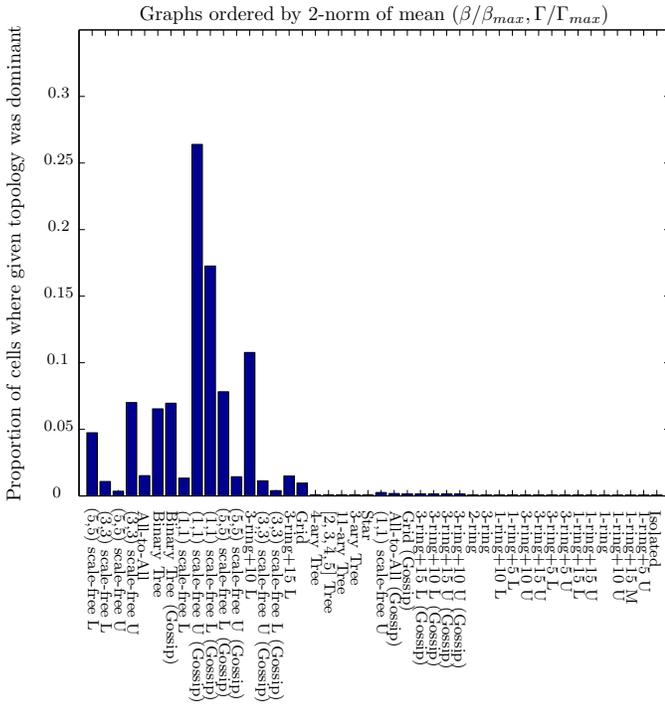


Fig. 10: Proportion of cells where a given topology is dominant. An 'L', 'M', or 'U' after a random graph indicates that it was selected because of its dominance amongst other random graphs of the same class in the lower, middle, or upper (Γ, β) plane. The '(Gossip)' designation indicates that the gossip protocol was used instead of the original simulcast model.

method, a dominance plot in the style of Figure 9 is not shown here. Instead, Figure 10 shows the distribution of dominant topologies (split equally among co-dominant topologies for those (Γ, β) locations). The topologies are ordered by Euclidean norm of the mean (Γ, β) location of the locations that they were (co-)dominant in, normalized by the maximum Γ and β values considered. Thus, topologies on the left tend to dominate closer to the origin than topologies on the right.

The results indicate that the gossip method tends to result in new regions of dominance that are closer to the upper-right of the decay-complexity plane than the regions of dominance that the topology using the original communication method was dominant in. In particular, the binary tree using gossip has dominant regions in the portion of the plane that the original binary tree dominated. Since the leaf nodes are the sensors for a binary tree, gossip and simulcast simulations begin with the same steps, so their overlap is not entirely unexpected. Another change of note between the two simulations is the amount of area gained by the (1,1) scale-free L that had very little of the area in the original method, but gains dominance in the regions where the binary tree and the (1,1) scale-free U were dominant. Additionally, the (1,1) scale-free U became dominant in many of the places that (1,1) scale-free U was dominant in with the original communication method. The (3,3) and (5,5) scale-free graphs also gain some dominance

in some of the more entropic regions of the plane. Overall, it would appear the gossip protocol adds some versatility in in the scale-free graphs by allowing to take some advantage of the low graph diameter without the bottlenecks observed at the super-nodes [22] in the original communication paradigm.

F. Topology Optimization

In this section, we discuss the results of a genetic algorithm (GA) [26] designed to directly optimize topologies in the above information flow simulation (as opposed to [7] where a number of sample topologies were simply compared). The goal of the optimization is to find a topology that performs better than the original set of topologies tested in [7], and then see if it then performs well in some neighborhood of that optimization point. The genome used is the upper half of the adjacency matrix (since the matrices are assumed symmetric) treated as a binary string. The fitness (objective) function used is the processed information volume of the candidate topology normalized by the processed information volume of the dominant topology from Figure 9 at that point.

A representative set of points for optimization are shown in Table I, along with the dominant topology at that point from Figure 9, and the normalized increase in processed information volume at that point (i.e., the fitness function). As can be seen from the fitness column, these optimizations are successful in increasing the processed information volume at the point being optimized. Dominance plots involving these optimized topologies indicate that they improve over the baseline topologies of [7] in a neighborhood of the point being optimized (see Figure 11). In some cases, the optimized topology completely dominates the original dominant topology in the original topologies dominant region, thus removing that topology from the dominance plot.

| # | Decay, Γ | Complexity, β | Orig. Region | Max Fit |
|---|-----------------|---------------------|--------------|----------|
| 1 | 0.051 | 0.75 | 15 | 0.111 |
| 2 | 0.199 | 1.25 | 14 | 0.693 |
| 3 | 0.121 | 4.25 | 13 | 0.578 |
| 4 | 0.399 | 2.00 | 3 | 0.0619 |
| 5 | 0.601 | 5.00 | 24 | 0.000435 |
| 6 | 0.601 | 4.75 | M5 | 0.000756 |
| 7 | 0.851 | 8.00 | M13 | 5.9e-10 |

TABLE I: Table of points in the decay-complexity plane that were chosen for optimization, the original dominant region corresponding to that point, and the maximum fitness obtained by the GA.

Optimization over multiple points simultaneously is also considered here. For technical reasons, the fitness function used for the multipoint optimizations is different than in the single point case. The multipoint fitness function is the sum of the normalized increases in processed information volume at each point, if that point had an increase, otherwise it was the decrease, but unnormalized. Formally, let $f(\mathcal{G}, \mathcal{S}, \Gamma, \beta)$ be the peak processed information volume for the graph \mathcal{G} , sensor set \mathcal{S} , geometric decay rate Γ , and linear processing complexity

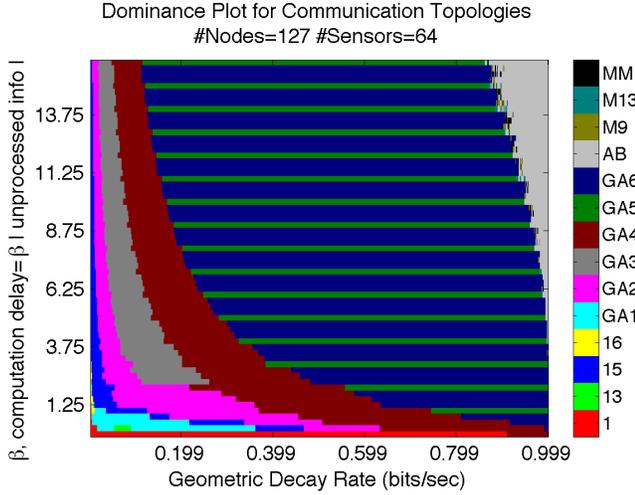


Fig. 11: (color online). Dominant topologies when single point optimized topologies are considered. Topologies 1,13,15,16, AB are the same as in Figure 8 and 9. Topologies GA_c for $c = 1, \dots, 7$ refers to the topology generated by optimizing at point c in Table I. M9 is co-dominated by the two (1,1) scale-free topologies from the original set, GA5 and GA6. M13 is co-dominated by topologies GA5 and GA7. MM is the combination of twelve co-dominated regions totaling less than 0.12% of decay-complexity pairs tested. AB-all topologies performed equally poorly.

factor β , according to information flow simulation described in Section III-C. Additionally, let $\bar{f}(\Gamma, \beta)$ be the maximum peak processed information volume with geometric decay rate Γ and linear processing complexity delay β over the original set of topologies considered in [7]. Next, define

$$g(\mathcal{G}, \mathcal{S}, \Gamma, \beta) = \begin{cases} f(\mathcal{G}, \mathcal{S}, \Gamma, \beta) - \bar{f}(\Gamma, \beta) & \text{if } f(\mathcal{G}, \mathcal{S}, \Gamma, \beta) \leq \bar{f}(\Gamma, \beta), \\ \frac{f(\mathcal{G}, \mathcal{S}, \Gamma, \beta) - \bar{f}(\Gamma, \beta)}{\bar{f}(\Gamma, \beta)} & \text{if } f(\mathcal{G}, \mathcal{S}, \Gamma, \beta) > \bar{f}(\Gamma, \beta). \end{cases}$$

Next given a set of points $\Phi = \{(\Gamma_i, \beta_i)\}$ in the decay-complexity plane, define the fitness function F by

$$F(\mathcal{G}, \mathcal{S}, \Phi) = \sum_{i=1}^{|\Phi|} g(\mathcal{G}, \mathcal{S}, \Gamma_i, \beta_i).$$

The fitness function F tends to drive the GA towards solutions that produce increases over all points in the optimization, rather than using the gains at one point to cover for the losses at another. Table II shows the results of this multipoint optimization. Here, the two-digit numbers correspond to optimizations at two points from Table I by concatenating the point numbers from the first column together. Similarly, the three-digit numbers in Table II correspond to three point optimizations from Table I.

The results of the 63 multipoint optimizations shown in Table II can be grouped into four ranges based on the value obtained by the fitness function. These are the positive fitnesses, the slightly negative fitnesses (fitnesses between 0 and -0.01),

| # | Fit | # | Fit | # | Fit | # | Fit |
|----|--------|-----|--------|-----|--------|-----|--------|
| 12 | 0.29 | 37 | 0.66 | 136 | -931.6 | 247 | -18.98 |
| 13 | -339.8 | 45 | 0.05 | 137 | -589.4 | 256 | 0.75 |
| 14 | -12.92 | 46 | 0.03 | 145 | -17.03 | 257 | 0.69 |
| 15 | 0.09 | 47 | 0.05 | 146 | -17.03 | 267 | 0.64 |
| 16 | 0.16 | 56 | -0.00 | 147 | -19.49 | 345 | -19.12 |
| 17 | 0.14 | 57 | 0.00 | 156 | 0.08 | 346 | -19.09 |
| 23 | 0.97 | 67 | 0.00 | 157 | 0.11 | 347 | -18.92 |
| 24 | -12.31 | 123 | -540.6 | 167 | 0.17 | 356 | 0.55 |
| 25 | 0.77 | 124 | -19.30 | 234 | -18.49 | 357 | 0.55 |
| 26 | 0.79 | 125 | 0.36 | 235 | 1.04 | 367 | 0.54 |
| 27 | 0.71 | 126 | 0.31 | 236 | 1.06 | 456 | 0.02 |
| 34 | -12.52 | 127 | 0.20 | 237 | 1.04 | 457 | 0.07 |
| 35 | 0.56 | 134 | -379.0 | 245 | -18.90 | 467 | 0.02 |
| 36 | 0.53 | 135 | -547.9 | 246 | -16.61 | 567 | -0.00 |

TABLE II: Fitness values for different two- and three-point optimizations. Two digit topology numbers correspond to two-point optimizations at the points in Table I, with each digit indicating a point of optimization. Similarly, three digit topology numbers correspond to three-point optimization.

moderately negative fitnesses (fitnesses between -12 and 20), and very negative fitnesses (fitnesses less than -100). Of the 63 total topologies, 41 were positive. As was the case with the single point optimizations, these multipoint optimizations with positive fitness increase performance over the baseline topologies in a neighborhood of the points of interest. Since these points are trying to create gains in fitnesses at multiple points, they do not perform as well at the original seven single point of optimizations. However, many of these multipoint optimized topologies do appear in the dominance plot (which is highly, highly, fragmented). In particular, with the addition of these multipoint optimized topologies, the original set of topologies is completely dominated, except for the fully connected graph which continues to dominate when there is no computational delay. This indicates that the multipoint optimizations are able to blend some of the characteristics of successful graphs at different points and find a topology that works well in a large range.

This wide-ranged improvement does not appear to always be possible, however, as some multipoint optimizations are negative. Despite positive fitnesses for many of the multipoint optimizations, 22 were negative and fell into one of the three ranges of negative fitnesses outlined above. The two fitnesses in the slightly negative range come from points 56 and 567. These two points are both multi-point problems containing the two points chosen for comparison in the newly discovered striated region of Figure 9. The other multipoint optimizations that contain both points 5 and 6 may have found topologies that result in an improvement in fitness at both points 5 and 6, but this seems unlikely. The processed information volumes at points 5 and 6 are much smaller than the processed information volumes at points 1-4, due to the increased effects of entropic drag. Thus, it seems more likely that optimizations at points 1-4 and both 5 and 6 are able to offset the negative fitness results with a greater increase at the third point. This is further supported by the fact that 567 has a negative fitness. Since point 7 is subject to even more computational delay

and entropic drag than points 5 and 6, gains in processed information volume at this point will be very small and unable to offset the negative values at points 5 and 6. Despite the fact that it was not possible to show improvement at both points five and 6, there was slight improvement in fitness from the original set of topologies at these points. Since the (1,1) scale-free graphs already performed well in both regions, the optimized topology at point 56 operates even better.

Of the 22 negative fitnesses, six of the multipoint optimizations are in the very negative range. All six of these multipoint optimizations contained both points 1 and 3. This indicates that the behavior in these regions is rich enough that it is impossible (or at least very unlikely) that a topology can create a positive fitness at both points simultaneously. Fourteen topologies are in the moderately negative region, and these are all the combinations containing both point 4 and one or more of points 1, 2, and 3, with the exception of combination 134, which is in the very negative region since it contains both points 1 and 3.

Point 4 presents an interesting case that merits additional discussion. In the baseline set of topologies this point was in the dominant region of the binary tree, serving as a boundary between the scale-free topologies on one side, and the striated region on the other, where (1,1) scale-free topologies also performed well. The results in Table II indicate that the optimization routine was unable to jointly optimize point 4 with one or more of points 1, 2, and 3 (which were in regions originally dominated by scale-free graphs). However, these three points are able to be jointly optimized with points 5, 6, and 7, which are points originally in the banded region (provided of course points 1 and 3 are not being jointly optimized, as discussed above). Thus, it seems possible to achieve a balance between the information-theoretic characteristics of the first three points with those of the last three points, but there is something about the characteristics of point 4 that make it difficult to optimize jointly with the first three points.

Since the full set of baseline and optimized topologies results in such a large number of different regions in the decay-complexity plane, the question of covering the decay-complexity plane with as few topologies as possible was investigated. The first step was to choose a proper metric to assess how well a given set of topologies performed. The metric used here is the mean ratio between processed information volume at each point for a set of topologies and the maximum achieved processed information volume over the entire set of baseline and optimized topologies. The next step in the covering process was to search exhaustively over all combinations of a fixed size from the set of candidate topologies to generate the first few sets of topologies that maximized the mean proportion of processed information volume. This is another combinatorial optimization problem for which only the first few iterations can be solved in a reasonable amount of time. Covering solutions using up to four total topologies are shown in Table III. Also shown in Table III are a number of sample coverings using various other sets of topologies, for comparison. The covering results

highlight the versatility of the three-point optimized topologies in their ability to perform well over a large region in the decay-complexity plane. In fact, a single three-point combination is able to outperform the original baseline set of topologies. These results also highlight the importance of considering topologies with negative fitness in the covering problem, as combinations 234 and 347 produced multi-point optimized topologies that had negative fitness values.

| Graph(s) | % Max Proc. IV |
|---------------------------|----------------|
| Isolated | 0.8439 |
| Fully Connected | 0.8630 |
| (3,3) scale-free (13) | 0.9096 |
| Original 31 | 0.9362 |
| Orig 31, one pt | 0.9772 |
| Orig 31, one & two pt | 0.9863 |
| 356 | 0.9444 |
| 234,457 | 0.9742 |
| Fully Conn., 234, 457 | 0.9846 |
| Fully Conn., 27, 347, 457 | 0.9908 |

TABLE III: Table of percentage of maximum processed information volume for a given set of graphs. The top four rows are graphs from the original set of 31 for comparison. The bottom four rows are the optimal coverings by one, two, three, and four graphs.

Overall, these results are encouraging for the development of agile C2 controllers that use optimal (or at least better) communication topologies and are able to change the topology on the fly. In [7], it was posited that if one could accurately estimate the information-theoretic characteristics of the C2 system, a good topology could be selected from the list of 31 topologies tested there. The results of the optimization routine demonstrate that one can actively optimize around a nominal operating point in the information-theoretic parameter space. This optimization appears to perform well in a region near this operating point, meaning that the optimizations should be robust to uncertainty and error in the estimation of the information-theoretic parameters. Additionally, it was shown that it is possible to achieve improvement over the baseline 31 topologies using very few of these optimized topologies, meaning that these could be used as a look-up table instead of optimizing around the nominal operating point.

IV. CONCLUSION

Here, the application of information theory to C2 was further developed. In particular, the quantification of the information lost to entropic drag is calculated for a number of sample scenarios. While these problems are far from complex, they are a special case of the more general (hidden) Markov model, and methods from there should prove useful in the information-theoretic characterization of more complex systems. Once this thread of research is completed, efficient estimators for entropic drag can be combined with agile control structures [8] to change the C2 topology on the fly in order to optimize information volume, and thus situational awareness. Additionally, these agile control structures should

be able adjust the abstraction level to further optimize the information volume in the C2 system.

With regards to the information flow simulation work in [7], here it was shown that the relationship between the information-theoretic characteristics of the system being observed and the relative performance of different communication topologies is more complex than previously demonstrated. By looking at cases where the computational delay are not integers, new regions of dominance were found. Interestingly, the dominant topologies in these regions were of the scale-free variety, which were previously only dominant in regions with low computational delay and entropic drag. This indicates that scale-free hierarchies (i.e., (1,1) scale-free graphs) have a combination of features that allows them to operate well in such disparate regions. This notion is supported by the results of the optimization algorithm. The optimization algorithm was able to jointly optimize topologies in the scale-free dominant regions and these regions of high computational delay and entropic drag, but not in the intermediate region originally dominated by the binary tree. The introduction of the gossip algorithm into the information flow simulation adds another facet to the problem of optimizing C2 topologies, as it is clear from these results that the communications protocol also has an affect on the dominance of a particular topology. In particular, the gossip protocol appears to increase the relative performance of scale-free graphs in regions of higher computational delay and entropic drag.

Overall, these investigations are important steps in the development of an agile C2 control system whose goal is to maximize situational awareness in real-time by managing the communication topology, communication protocol, and abstraction level. Additional tools needed to achieve this goal include information-theoretic characterization of C2 systems, efficient state estimation and prediction, active diagnosis, the control routines themselves, etc. Since the amount of information that can be obtained by an actuator has a very specific relationship on the efficacy of control that can be imposed by that actuator [5], [6], we hypothesize that these concepts are important in the broader context of control problems, as well.

ACKNOWLEDGMENTS

This work was funded in part by ONR contract N00014-0-C-0396.

REFERENCES

[1] A. Cebrowski and J. Garstka, "Network-centric warfare: Its origin and future," in *US Naval Institute Proceedings*, vol. 124, no. 1, 1998, pp. 28–35.

[2] D. S. Alberts, J. J. Garstka, and F. P. Stein, *Network Centric Warfare*. CCRP Publication Series, 1999.

[3] D. S. Alberts, J. J. Garstka, R. E. Hayes, and D. T. Signori, *Understanding Information Age Warfare*. CCRP Publication Series, 2001.

[4] C. E. Shannon, "A mathematical theory of communications," *The Bell Systems Technical Journal*, vol. 27, pp. 379–423, 623–656, July, Oct 1948.

[5] H. Touchette and S. Lloyd, "Information-theoretic limits of control," *Phys. Rev. Lett.*, vol. 84, no. 6, pp. 1156–1159, Feb 2000.

[6] —, "Information-theoretic approach to the study of control systems," *Physica A: Statistical Mechanics and its Applications*, vol. 331, no. 1-2, pp. 140 – 172, 2004.

[7] D. Scheidt and K. Schultz, "On optimizing command and control structures," in *Proceedings of the International Command and Control Research and Technology Symposium, 2011*, 2011.

[8] K. Schultz, "Towards agile control of ship auxiliary systems," in *Resilient Control Systems (ISRCs), 2011 4th International Symposium on*, aug. 2011, pp. 154 –157.

[9] J. Boyd, "The essence of winning and losing," *Unpublished lecture notes*, 1996, available online http://pogoarchives.org/m/dni/john_boyd_compendium/essence_of_winning_losing.pdf.

[10] A. Gelb, *Applied optimal estimation*. MIT press, 1999.

[11] P. Bougerol, "Kalman filtering with random coefficients and contractions," *SIAM Journal on Control and Optimization*, vol. 31, p. 942, 1993.

[12] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1453–1464, 2004.

[13] A. Censi, "Geometric remarks on Kalman filtering with intermittent observations," *Arxiv preprint arXiv:0906.1637*, 2009.

[14] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 39, no. 4, pp. 1152–1178, oct. 2003.

[15] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4091–4104, nov. 2006.

[16] R. Sukhvasi and B. Hassibi, "The Kalman like particle filter: Optimal estimation with quantized innovations/measurements," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE, 2009, pp. 4446–4451.

[17] D. Hernando, V. Crespi, and G. Cybenko, "Efficient computation of the hidden Markov model entropy for a given observation sequence," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2681–2685, 2005.

[18] D. Scheidt, A. Watkins, and M. Miller, "Information theory, coordination & control in real-time anti-torpedo engagements," in *Submarine Technology Symposium*, Laurel, MD, 2008.

[19] D. Scheidt and M. Pekala, "The impact of entropic drag on command and control," in *The 12th Annual International Command and Control Research and Technology Symposium*, 2007.

[20] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[21] M. E. J. Newman and D. J. Watts, "Renormalization group analysis of the small-world network model," *Physics Letters A*, vol. 263, no. 4-6, pp. 341–346, 12 1999.

[22] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[23] C. G. Cassandras and S. LaFortune, *Introduction to Discrete Event Systems*, 2nd ed. New York: Springer, 2007.

[24] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[25] Z. Haas, J. Halpern, and L. Li, "Gossip-based ad hoc routing," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1707–1716.

[26] M. Mitchell, *An introduction to genetic algorithms*. The MIT press, 1998.