Distributed Information Fusion in Communications Networks with Ad Hoc Connectivity and Non-Deterministic Link Characteristics

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Sciences in Systems Engineering at George Mason University

By

Todd W Martin
Bachelor of Sciences
George Mason University, 2005

Director: Kuo-Chu Chang, Professor
Department of Systems Engineering and Operations Research

Fall Semester 2005
George Mason University
Fairfax, VA

DEDICATION

To Helen, Rachel, and Andrew.

ACKNOWLEDGEMENTS

I would like to thank the following outstanding individuals and organizations for their help in making this effort possible: Dr. K.C. Chang for his guidance and advice throughout this research effort; Dr. Chun-Hung Chen and Dr. John Shortle for their participation in the thesis committee; Preston Marshall (currently of DARPA) for providing the final stimulus to pursue the Master's program; Dr. Paul Kolodzy for his advice and encouragement; and Science & Technology Associates Inc. for their support and flexibility. I must also recognize my parents for their encouragement and support throughout all my endeavors; and most of all, my wife Helen and children Rachel and Andrew for their infinite patience and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

ABSTRACT

DISTRIBUTED INFORMATION FUSION IN COMMUNICATIONS NETWORKS WITH AD HOC CONNECTIVITY AND NON-DETERMINISTIC LINK CHARACTERISTICS

Todd Martin

George Mason University, 2005

Thesis Director: Dr. Kuo-Chu Chang

This thesis establishes algorithms and analytical methods for implementing and evaluating distributed information fusion in networks with non-deterministic communications connectivity. The methods developed in this thesis encompass any sequence of fusion events within an arbitrary network resulting from random channel characteristics, network delays, and ad hoc networking.

A distributed fusion approach is developed and proposed as a general solution for distributed fusion agents, enabling each agent to operate autonomously and collaboratively as network conditions allow. The resulting Local Fusion Graph method enables fusion agents to exchange data on an ad hoc or opportunistic basis for distributed fusion. The method's decentralized approach is inherently able to overcome difficulties experienced by existing distributed fusion methods resulting from messages that are dropped, delayed, or received out of order. The method provides an algorithm that can be

implemented into distributed fusion agents without a priori knowledge of network architecture, membership, or communications patterns. It also provides a graphical approach that can be used for analysis as well as simulation model development.

A stochastic-based fusion formulation is similarly developed and proposed as a general solution for distributed estimation and trend analysis. The method encapsulates the effects of non-deterministic behaviors and characteristics into probabilistic factors that are integrated into the fusion equations. The resulting stochastic fusion method enables average estimation performance of distributed fusion networks having non-deterministic characteristics and ad hoc connectivity. The method also greatly simplifies the simulation and analysis performance approximations for distributed fusion by providing significant reductions in simulation complexity and computational requirements while using non-idealized communications characteristics.

The two methods are implemented in computer-based models for distributed tracking in networks with non-deterministic connectivity. The results of the computer models are used to assess general trends in estimation capabilities with respect to average network connectivity. The analyses also address communications requirements relative to other distributed fusion approaches in the context of estimation accuracy.

The results of this thesis are the ability to implement, model, and assess distributed information fusion in arbitrary communications networks with realistic communications characterizations. The solutions proposed in this thesis demonstrate increased estimation capabilities under non-ideal networking conditions that are characteristic of wireless mobile networking environments.

1 INTRODUCTION

Distributed information and data fusion provide core capabilities to numerous applications such as robotics, inference and reasoning, situational awareness, medical diagnostics, navigation, and tracking. The increasing capabilities and ubiquity of computing technologies and data networks continue to advance information exploitation abilities for combining data from multiple remote sources to produce increased value. In particular, information fusion is a key enabler for the United States Department of Defense (US DoD) network-centric Command, Control, Communications and Intelligence (C^3I) capabilities that are under development [1].

The DoD Network Centric Warfare (NCW) concept combines the operational aspects of distributed operations and information processing with emerging networking and communications technologies [2-5]. NCW utilizes common situational awareness that is developed by integrating shared information across geographically-dispersed forces, sensors, platforms, and decision aids to achieve intended objectives [3,6]. The concept aims to utilize every entity – whether a solder, weapon, or satellite – as an information source and simultaneously decrease information volume via local processing of the data sources to achieve actionable results. These results can then be passed throughout the network to other entities for which the information is relevant [3].

Because of the high degree of mobility and highly-distributed nature of tactical operations, most future military information fusion applications will be hosted on wireless tactical networks. While all communications networks are inherently probabilistic in nature, mobile wireless tactical networks rarely provide the same degree of connectivity as wired ones. For wireless networks, signal propagation becomes a primary factor in determining network connectivity. Signal propagation is affected by a number of random processes including signal attenuation, deflection, and fading as well as unwanted noise from sources causing intentional and unintentional interference [7-10]. The wireless signal propagation effects are combined with network behaviors that are characterized by stochastic concepts such as delay and queuing theory [11].

In addition to these stochastic factors, mobile wireless tactical networks must contend with non-deterministic factors – and perhaps non-predictable factors – that arise from implementation. For many of the mobile tactical platforms, communications capabilities are supporting functions that enable the primary operations of the platform. As such, communications capabilities are ideally determined by an organization's operations and will not be the primary planning factor or constraining capability. Thus wireless networks and their applications must be adaptive to the inherent uncertainty that results from operational implementations. This added degree of uncertainty is especially pronounced for ground-based communications, which must adapt to the combined effects of mobility, propagation limitations, and terrain. Of particular note are communications conditions in urban operations where buildings and background noise are significant nuisances for wireless communications [12].

Reliable communications for such uncertain military and commercial applications environments are being address via a number of communications and networking technology development areas. These efforts seek to eliminate RF spectrum, network functionality, network disruptions, and dynamic mobility as fundamental constraints on tactical wireless communications capabilities [13-26]. The trend is therefore away from highly robust but inflexible communications systems to highly-tolerant and adaptive networks of flexible communications systems [27]. Rather than engineer radios and networks that produce highly-reliable connectivity but constrain operational implementations, new technologies are creating the ability to operate in the presence of the frequent disruptions and uncertainty created by unpredictable operational implementations.

Chief among these technologies are those under development for enabling mobile ad hoc networking (MANET). A number of recent demonstrations and simulation studies provide valuable insight into future military tactical MANET network characteristics. The first of these studies is an assessment of the Defense Advanced Research Projects Agency (DARPA) Small Unit Operations Situational Awareness System (SUO SAS) Program. The SUO SAS Program developed MANET-based technologies that furthered the ability to establish reliable and flexible communications in restrictive tactical environments [17,19,21-23]. Program demonstration results reported successful achievement of the program's goals for voice, data, and geolocation capabilities, but also revealed average network connectivity levels ranging from 73% to 81% in the test environment [17].

Another demonstration of MANET capabilities is found in the DARPA Future Combat System (FCS) scalable mobile network demonstrations [26]. Like the SUO SAS demonstrations, the FCS demonstration focused on the assessment of communications connectivity, quality of service, and robustness of a military MANET in an operational environment that was representative of realistic tactical implementations. The demonstration results presented in Table 1 and Figure 1 show that message delivery rates (MDR) in military tactical MANETS connectivity fluctuate widely with time and rarely provide communications nodes with full connectivity. Subnets with shorter link ranges fared better than longer-range inter-subnet connections but still experienced significant network disruptions. In general, the rapid fluctuations in network connectivity are attributed to the confluence of factors such as the relative mobility of nodes, communications range, propagation changes, and terrain masking which are not generally predictable [26].

Table 1: Message Delivery Rate Statistics Summary for the FCS Scalable Mobile Network Demonstration [26].

Network Type	**Max. MDR**	**Min. MDR**	**Avg. MDR**	**MDR Std. Dev.**
Intra-Subnet	100%	69%	95%	11
Inter-Subnet	100%	0%	68%	25
Total Network	96%	74%	85%	8

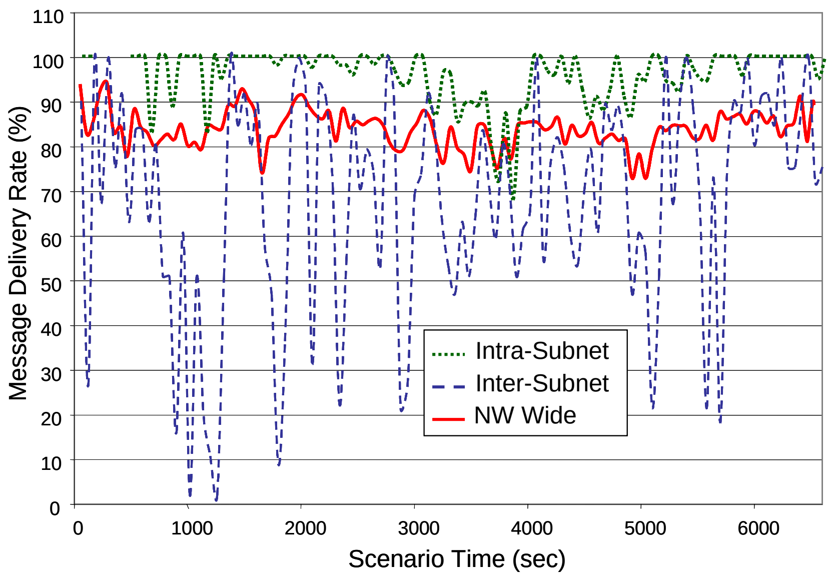

Figure 1: Message Delivery Rates vs. Time for the FCS Scalable Mobile Network Demonstration [26].

Comparable results are found in simulation-based studies of MANET capabilities. In simulations performed for [28], the FCS demonstration data was used as a basis for examining the performance of various network protocols in the FCS MANET environment. Average packet delivery success rates range from 55% to 93% using a range of networking protocols with delivery times ranging across three orders of magnitude from 5 msec to over 10 s. The study demonstrated a high variability of performance across communications nodes in the scenario: some nodes experienced 90% message loss while others maintained near-perfect connectivity. With similar performance results for each of the protocols examined, the unpredictable variability is once again attributed to the combination of in situ conditions including range, propagation, and line-of-site with other nodes.

The results of the FCS studies are consistent with those presented in [29]. This study simulated MANET operations in a military tactical environment, and produced results showing message delivery rates of 79% to 97% for voice-based traffic and 81% to 91% for IP-based traffic. Of particular note is that the model did not implement transmission security (TRANSEC), network security (NETSEC), or communications security (COMSEC) elements [29]. The associated overheads for these security features would create additional delays and perhaps further reduced delivery rates.

Despite the stark network connectivity achievable by state-of-the-art MANET technologies, each of the referenced studies demonstrated the ability to run various distributed applications across the networks [26,28,29]. These applications were designed to handle the frequent disruptions and delays experienced in tactical MANET environments. Had they been designed otherwise, they surely would not have been able to function.

Given the results of these programs and studies, it is clear that distributed information fusion applications hosted on these networks must be able to operate in the presence of significant and non-deterministic disruptions and delays. Maintaining fusion capabilities in the presence of disruptions and delays, however, still remains a challenge [30,31]. Furthermore, the inclusion of non-ideal networking conditions is largely absent from core information fusion texts [32-35]. The texts will occasionally acknowledge uncertainty in distributed fusion communications, but the formulations assume a priori knowledge of communications architectures and certainty of fusion and communications events. As such these fusion methods do not adequately address the uncertain nature of information fusion in wireless networks.

Thus there is a need to develop a general approach to distributed information fusion in networks with non-deterministic communications characteristics that are unknown a priori. The method must be valid for networks with delays and disruptions brought about by a number of different factors such as mobility, dynamic membership of networks, message delays, dynamic routing, and realistic communications channel characteristics.

The following chapters develop and evaluate two distributed fusion methods that incorporate the uncertainties introduced by wireless networking delays, disruptions, and uncertain connectivity. Chapter 2 surveys existing distributed fusion approaches and develops a distributed fusion method that can be directly implemented in distributed fusion agents connected by any arbitrary communications network. Chapter 3 describes the incorporation of the resulting algorithms into a computer-based model and presents an analysis of simulation results. Chapter 4 develops a stochastic-based distributed fusion formulation that simplifies analyses of average estimation capabilities. A simplified form is developed for ad hoc networks and the results of a computer-based implementation are analyzed in light of the results from Chapter 3.

2 DISTRIBUTED FUSION ALGORITHM FOR NETWORKS WITH NON-DETERMINISTIC CONNECTIVITY

The fundamental purpose of this chapter is to develop the essential concepts and algorithms that enable fusion agents to operate in a distributed manner with ad hoc interactions and non-deterministic communications channels. These algorithms do not require a priori knowledge of communications patterns or data flow, nor do they require a priori knowledge of fusion agent interactions and fusion events. They also seek to limit the amount of data communicated between agents as well as the processing required of each agent. The resulting algorithms demonstrate consistency with known information fusion fundamentals and known formulations for deterministic connectivity.

The discussion begins with a description of the fusion fundamentals that are foundational to the approach that is derived herein, followed by an assessment of existing approaches to distributed fusion in the presence of imperfect communications. A distributed fusion methodology that is both analytically tractable and can be readily implemented in distributed fusion agents is derived and proposed as a general solution for distributed fusion systems. The chapter closes out with a discussion of observations and considerations pertaining to the proposed solution and leads into the simulation and analysis presented in the next chapter.

2.1 Relevant Fusion Fundamentals

The theoretic fundamentals of distributed information fusion are well documented and have been studied in depth. Of particular significance to the work presented here are [36-41]. The theories developed in those works are independent of information flow patterns, including non-deterministic patterns. It is noted, however, that practical applications of these theoretical results to non-deterministic information flows has remained a challenge. The need to identify and remove common information from the data sets to be fused while minimizing the amount of data exchanged between agents is the primary difficulty.

The basic fusion process as described in [36] follows from set theory, where the combination of n event probabilities $\Phi(I_i)$ can be represented as:

$$\Phi\left(\bigcup_{i=1}^{n} I_i\right) = \sum_{i=1}^{n} (-1)^{i+1} S_i \tag{1}$$

where the S_i terms are the combinations of i event probabilities such that $S_1 = \sum_{i=1}^{n} \Phi(I_i)$, $S_2 = \sum_{i=1, j=i+1}^{n-1} \Phi(I_i \cap I_j), \ldots,$ $S_n = \Phi(I_1 \cap I_2 \cap \ldots I_n)$ [42-43]. The alternating addition and subtraction of joint probabilities from Equation (1) removes conditional dependencies from the data sets, which is in the form of common information. In [36] and [37], the application of this elementary property to the fusion of linear Gaussian state estimate and covariance ($\hat{\mathbf{x}}(k)$ and $\mathbf{P}(k)$, respectively) at time k gives the following fusion equations:

$$\hat{\mathbf{x}}_j(k) = \mathbf{P}_j(k)\left[\left(\sum_{i=1}^{n}\mathbf{P}_i^{-1}(k \mid k)\hat{x}_i(k \mid k)\right) - \overline{\mathbf{P}}_j^{-1}(k)\overline{\mathbf{x}}_j(k)\right] \tag{2}$$

$$\mathbf{P}_j(k) = \left[\left(\sum_{i=1}^{n}\mathbf{P}_i^{-1}(k \mid k)\right) - \overline{\mathbf{P}}_j^{-1}(k)\right]^{-1} \tag{3}$$

where the following conventions are used:

$\hat{\mathbf{x}}_j(k)$ and $\mathbf{P}_j(k)$ are the fused state (vector) estimate and covariance (matrix) for agent *j* at time *k*;

$\hat{\mathbf{x}}_i(k \mid k)$ and $\mathbf{P}_i(k|k)$ are the local state estimate and covariance for agent *i* at time *k*;

$\overline{\mathbf{x}}_j(k)$ and $\overline{\mathbf{P}}_j(k)$ are the time-updated state estimate and covariance from prior times that are shared among the various combinations of *n* information sets being fused at time *k* by agent *j*.

While the removal of duplicative information is straightforward in the theoretical formulation, identification of the duplicative information for distributed estimation systems can be difficult in arbitrary implementations. The difficulty is due to the need to know the values of the data sets as well as their interrelationships resulting from past fusion events.

The Information Graph technique presented in [36,37] provides an analytical tool for identifying duplicative information in distributed estimation systems. The approach is a symbolic representation of the collection, propagation, and fusing of data among a set of fusion agents. An example of an Information Graph is shown in Figure 2, where a cyclical communications pattern is demonstrated. Each numbered row of symbols represents the events of a given agent, where squares are sensing events, solid circles are transmission events, and open circles are fusion events. The hexagonal symbols represent the state of the observed object, and the horizontal sequencing of symbols represents time progression. Within each time step, each agent may perform time updates of estimates, receive sensor data, perform measurement updates of estimates, transmit the resulting local estimate to other agents, and fuse estimates received from other agents.

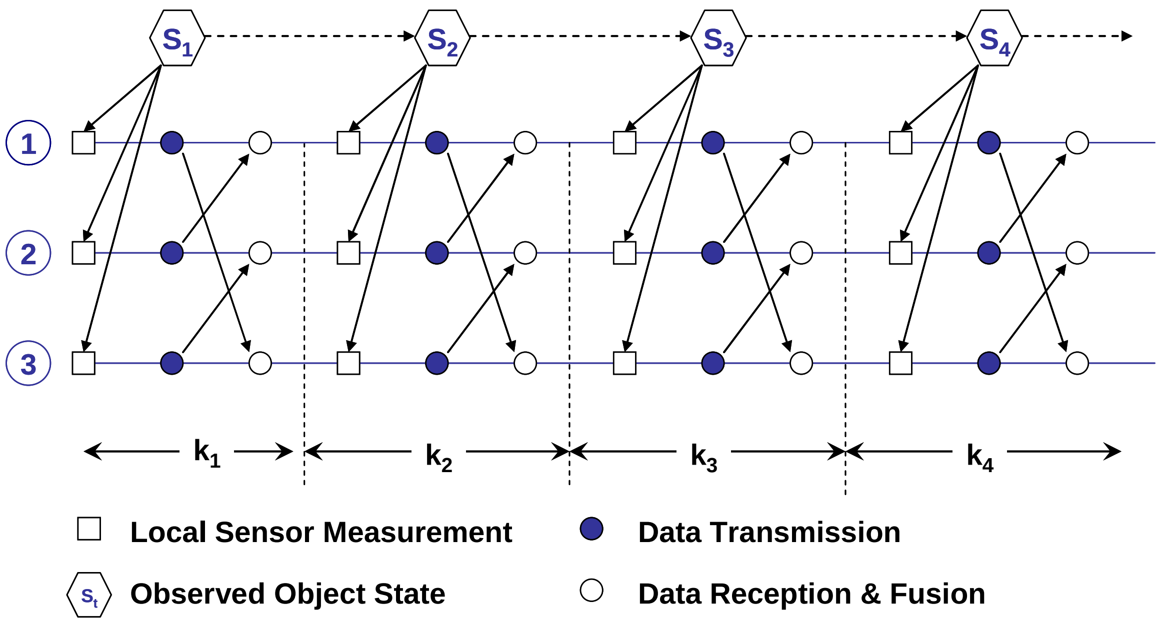

Figure 2: Information Graph of Cyclical Communications in a Distributed Estimation Network of Three Fusion Agents [36,37].

Common information is found by recursively tracing information flow paths in reverse from the fusion node of interest and locating nodes in the graph where two or more branches of these paths intersect. When common a node is found, the search for common elements terminates at that node while searches may continue along other branches that do not intersect at the common node. The process continues until the search paths terminate at common nodes or the time origin of the graph has been reached [36,37]. Representing the transmission nodes as a set of agent-time pairs given as (*agent_id*, *time*), the common information for agents 1 and 2 at time k_4 in Figure 2 is the set $\{(1,k_2), (2,k_3)\}$. Applying the set theoretic formulation in Equation 1:

$$\begin{aligned}\Phi\big(I_1(k_4)\big) &= \Phi\big(I_1(k_4,k_4)\cup I_2(k_4,k_4)\big)\\ &= \Phi\big(I_1(k_4,k_4)\big)+\Phi\big(I_2(k_4,k_4)\big)-\Phi\big(I_1(k_4,k_4)\cap I_2(k_4,k_4)\big)\end{aligned}. \tag{4}$$

From the example, the common elements are represented as:

$$\begin{aligned}\Phi\big(I_1(k_4,k_4)\cap I_2(k_4,k_4)\big) &= \Phi\big(I_1(k_2,k_2)\cup I_2(k_3,k_3)\big)_{k_4}\\ &= \Big\{\Phi\big(I_1(k_2,k_2)\big)_{k_3}+\Phi\big(I_2(k_3,k_3)\big)-\Phi\big(I_1(k_2,k_2)\cap I_2(k_3,k_3)\big)\Big\}_{k_4}\end{aligned} \tag{5}$$

where the subscripts k_3 for the first term and k_4 for the entire expression indicate time-update operations. The update of the information at $(2,k_2)$ is performed so that it can be combined with $(1,k_3)$, and the update of the entire expression must be performed such that the prior terms are at the same time reference as the two original information sets being fused.

The algorithm for properly translating prior sets is straightforward for single common information sets but is more complex for multiple common information sets, particularly if the sets have dissimilar time references. Thus the derivation is presented in Appendix A and the results are summarized here:

1) Single Common Prior: Valuation is found through the recursive application of the transition function. This function is given as the prediction function for estimation in linear dynamic systems.
2) Multiple Common Priors with Same Reference Frame: The common prior sets are first fused at their originating reference then recursively updated to the current reference frame.
3) Multiple Common Priors with Different Reference Frames: The common prior sets are translated to the reference frame of the most recent set and then fused before being recursively updated to the current fusion reference frame.

Thus information sets are fused and time-updated according to their order and pedigree of occurrence in the iterative equation that results from identifying prior information. As an example, consider the iterative set of equations for fusing three arbitrary covariance matrices is given by:

$$\begin{aligned}\Phi(I) = \bigcup_{i=1}^{3}\Phi(I_i) = \Phi(I_1) + \Phi(I_2) + \Phi(I_3) - \\ \{\Phi(I_1 \cap I_2) + \Phi(I_1 \cap I_3) + \Phi(I_2 \cap I_3)\} + \Phi(I_1 \cap I_2 \cap I_3)\end{aligned} \tag{6}$$

Proper fusion requires that the time indices for each set or term in Equation (6) are consistent with each other. Likewise, the time indices of the elements *within* a given conjunctive term must be consistent before they can be fused to produce the conjunctive result. Thus the elements of $\Phi(I_i \cap I_j ... \cap I_n)$ must all have identical time indices before they can be fused.

Thus returning to the example and Equation (5), the common information search process must be performed for {(1,k_2), (2,k_3)} to determine if they contain shared information. Conducting this search produces the set {(1,k_1), (2,k_1)} giving the following result:

$$
\begin{aligned}
\Phi\big(I_1(k_2,k_2)\cap I_2(k_3,k_3)\big) &= \Phi\big(I_1(k_1,k_1)\cup I_2(k_1,k_1)\big) \\
&= \Phi\big(I_1(k_1,k_1)\big)+\Phi\big(I_2(k_1,k_1)\big)=\Phi\big(I_1(k_1)\big)
\end{aligned}
\tag{7}
$$

By recognizing that $\Phi\big(I_1(k_1,k_1)\cup I_2(k_1,k_1)\big)$ is simply the fused result for agent 1 at time k_1, the final relationship becomes:

$$
\begin{aligned}
\Phi\big(I_1(k_4)\big) &= \Phi\big(I_1(k_4,k_4)\cup I_2(k_4,k_4)\big) \\
&= \Phi\big(I_1(k_4,k_4)\big)+\Phi\big(I_2(k_4,k_4)\big)- \\
&\quad \Big\{\Phi\big(I_1(k_2,k_2)\big)_{t_3}+\Phi\big(I_2(k_3,k_3)\big)-\Phi\big(I_1(k_1)\big)_{k_3}\Big\}_{k_4}
\end{aligned}
\tag{8}
$$

where the subscript k_4 applied to the second term (i.e. the common information from prior time steps) indicates a time update of the associated parenthetic results. Applying the fusion Equations (2) and (3) for Gaussian state estimation produces the following solution for fused result for agent 1 at time k_4:

$$
\hat{\mathbf{x}}_1(k_4)=\mathbf{P}_1(k_4)\left\{
\begin{array}{l}
\big(\mathbf{P}_1^{-1}(k_4\mid k_4)\hat{\mathbf{x}}_1(k_4\mid k_4)+\mathbf{P}_2^{-1}(k_4\mid k_4)\hat{\mathbf{x}}_2(k_4\mid k_4)\big)- \\
\left(\begin{array}{l}
\mathbf{P}_1^{-1}(k_2\mid k_2)_{t_3}\hat{\mathbf{x}}_1(k_2\mid k_2)_{k_3}+\mathbf{P}_2^{-1}(k_3\mid k_3)\hat{\mathbf{x}}_2(k_3\mid k_3)- \\
\mathbf{P}_1^{-1}(k_1)_{k_3}\hat{\mathbf{x}}_1(k_1)_{k_3}
\end{array}\right)_{k_4}
\end{array}
\right\}
\tag{9}
$$

$$
\mathbf{P}_1(k_4)=\Big\{\big(\mathbf{P}_1^{-1}(k_4\mid k_4)+\mathbf{P}_2^{-1}(k_4\mid k_4)\big)-\big(\mathbf{P}_1^{-1}(k_2\mid k_2)_{t_3}+\mathbf{P}_2^{-1}(k_3\mid t_3)-\mathbf{P}_1^{-1}(k_1)_{k_3}\big)_{k_4}\Big\}^{-1}
\tag{10}
$$

From this analytical perspective, the Information Graph provides the ability to identify common information elements and produce the needed formulations to ensure that the fused data contains no duplicative information. Furthermore, the Information Graph can be used with any arbitrary distributed fusion process. The realization of the approach in actual networks of distributed fusion nodes, however, is prohibitive because the Information Graph method is a centralized process that builds upon a global or omniscient view of events within the network rather than upon the perspective of the local agents. Individual fusion agents do not typically have the global perspective required for the Information Graph, and obtaining it from neighboring agents is costly in terms of communications resources.

2.2 Existing Distributed Information Fusion Methods

Two recent approaches have been proposed for allowing distributed agents to identify and remove common information in a communications-efficient manner. The works, however, focus on removing dependencies due to measurement correlations rather than due to correlations in communications histories. While the efforts are not comprehensive in scope, they do provide insight regarding information dependencies in distributed fusion.

The first approach presented in [44,45] utilizes a five-step process for filtering out duplicate information from remote estimates sent by other fusion agents. The approach combines the local measurement with measurements extracted from the incoming estimates using the technique described in [46]:

$$\mathbf{z}_i(j) = \mathbf{R}\left(\mathbf{H}^T\right)^{-1}\left(\mathbf{P}_i^{-1}(k \mid k)\mathbf{x}(k \mid k) - \mathbf{P}_i^{-1}(k \mid k-1)\mathbf{x}_i(k \mid k-1)\right) \tag{11}$$

where all model parameters at times k and k-1 are assumed to be known at the fusing agent. Once all the measurements from the other network participants are known, the agent calculates the fused estimate and propagates it back into the network. The technique is presented in the context of a fully-connected and decentralize sensor network with potentially random delays in message delivery between agents. This method is extended in [47], which utilizes evidence propagation techniques of Bayesian Networks as means for identifying common measurements data and calculating estimates within a fusion agent.

While the techniques in [44,45,47] address some issues associated with imperfect and non-deterministic communications, they are not broadly applicable to arbitrary network connectivity.3ee First and foremost, the method is limited to fully connected networks comprising a known set of fusion agents. Without knowing the participants in the network, a fusion agent is unable to fully identify and remove duplicate information from the estimation calculation. Second, the method does not address the concern of missed messages from other agents. As discussed previously, networks of wireless sensors and fusion agents will experience message losses due to link reliability. In addition to these two fundamental issues, other concerns affecting performance and scope of applicability exist but are not fully known:

1) The method may not be able to process estimates in networks with highly-variable and unpredictable message delay times. Since they must wait until all measurements

at a given time step have been found, fusion agents may need to sacrifice accuracy and fuse an incomplete measurements set or accept long time windows between fusion events.

2) The method may not be applicable to networks comprising a heterogeneous collection of fusion agents with very different rates of estimation transmission. Similarly, the differing model parameters and performance capabilities of the agents must be known a priori.
3) The method cannot process data received out of order if it has already been fused into the estimate at a prior time [45]. While out-of-order data is treated as an exception in the discussion, this type of data delivery is likely to be very common in MANETS [15-23,26,48].

A second distributed fusion method for the removal of data dependencies is presented in [39]. Like [47], the approach utilizes Bayesian Networks to establish independence in measurements data. It clearly delineates the removal of dependencies due to communications histories from dependencies due to non-deterministic state and measurement behaviors. The Information Graph is presented as a solution for managing communications-related data dependencies, and Bayesian Networks are developed as a means for managing measurements-related data dependencies. While use of the Information Graph for information management in distributed networks is proposed on a conceptual level, there is no development of its application in an tactical fashion beyond [36,37]. In both sources, the architecture-independent derivation ends with a presentation the fusion equations given by Equations (2) and (3), and then transitions to discussions of fixed communications structures among a set of known agents as implementation examples. Furthermore, the communications channels continue to be modeled with assured message delivery and no delay.

Given the survey results of efforts that address correlation between fusion data in distributed fusion networks, it is clear that an approach has yet to be fully developed that enables distributed fusion in networks with unpredictable communications patters. As such, reliable distributed information fusion in the presence of non-deterministic communications connectivity remains a challenge [30,31]. The Information Graph approach provides a theoretical basis for identifying and removing data correlations due to communications histories, but it has not been demonstrated beyond deterministic communications patterns. The efforts presented in [44,45,47] do address delays in communications networks, but the technique appears to break down in cases where data is received out of order. Furthermore, its applicability to wireless MANET-type networks is doubtful due to highly-dynamic network behaviors, rapidly-varying channel characteristics, and changing membership of MANETs.

2.3 Overview of the Local Fusion Graph Method

As discussed in the previous section, the goal of this effort is to develop a distributed fusion method that is able to remove dependencies in estimation data for any arbitrary communications pattern. The method must be applicable to distributed fusion in networks with non-deterministic characteristics that are unknown a priori. The uncertainty may be brought about by a number of different factors such as agent mobility, network dynamics, and communications channel characteristics. The method must also show that it holds to the theoretical fusion results and is consistent with results derived in deterministic cases.

The intent in developing the method – termed “Local Fusion Graph” method – is to impose a minimal number of assumptions and restrictions. It will be shown that the Local Fusion Graph method is applicable to any flow of information and associated fusion techniques. Thus it can be applied to networked agents with arbitrary connectivity and message delays as well as random or non-synchronous local sensing and communication rates. The method adheres foremost to the fundamental principles represented in Equations (1) through (3) and makes no known assumptions that restrict the extent of its applicability within the realm of distributed fusion networks.

As previously mentioned, one key principle is that the algorithms must be applicable to individual fusion agents. As a result, each fusion agent organizes the history of fusion events and associated information into the form of a graph containing sets of interconnected nodes. Similar to the Information Graph, the nodes of the graph represent information for a given agent-time pair. During a fusion event, the local agent receives the essential data from other agents, and grafts unique information into the local fusion graph, and identifies common information to be incorporated into the fusion equations. The result is a unique fusion graph for each agent containing non-duplicative information.

The algorithms for performing these actions are derived in the next section. An abbreviated derivation is also presented in [49], which provides a very early precursor of the Local Fusion Graph method. The algorithms described in [49], however, are used for building fusion trees rather than fusion graphs. While the general algorithms are consistent with those presented here the graphical representation has been refined to more closely represent the underlying algorithms and data relationships. Additionally, the derivation presented here is expanded to address a wider range of considerations and more fully represent the capabilities of the technique.

2.4 Derivation of the Local Fusion Graph Method

The derivation and explanation of how the local fusion graphs are developed and maintained are best done using an example. Consider the network of three distributed fusion agents shown in Figure 3. The communications links connecting them each comprise two characteristics of interest:

1) Message delivery or link probability, which is given as the probability that a message is transmitted by one agent and is received by another. This probability can be a function of mobility, physical channel characteristics such as noise and fading, receiver characteristics, and network performance [7-9,11,50].
2) Message delay, which is given as the elapsed time between message transmission at the source agent and reception at the receiving agent. The theoretical and practical foundations of network delay problems that result from factors such as network congestion, routing, and link quality are also well developed [11].

Each of these factors is characterized by uncertainties that may change as a function of time and may be unique to each transmitter-receiver pair.

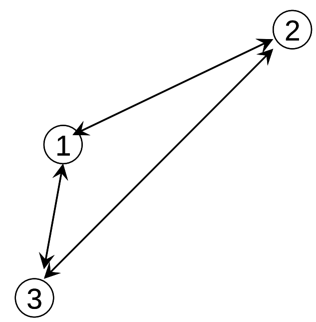

Figure 3: Example Network of Distributed Fusion Agents.

To generate a sample scenario for the derivation, the series of communications events across four consecutive time steps shown in Figure 4 is presented. The links shown between nodes are successfully achieved with some probability of success and delay. The solid links indicate successful message transmission and reception within the same time step, while dashed links indicate that the message transmission and reception occur in different time steps as indicated by the associated labels. The absence of a link indicates data was either not sent or that data was sent but not received.

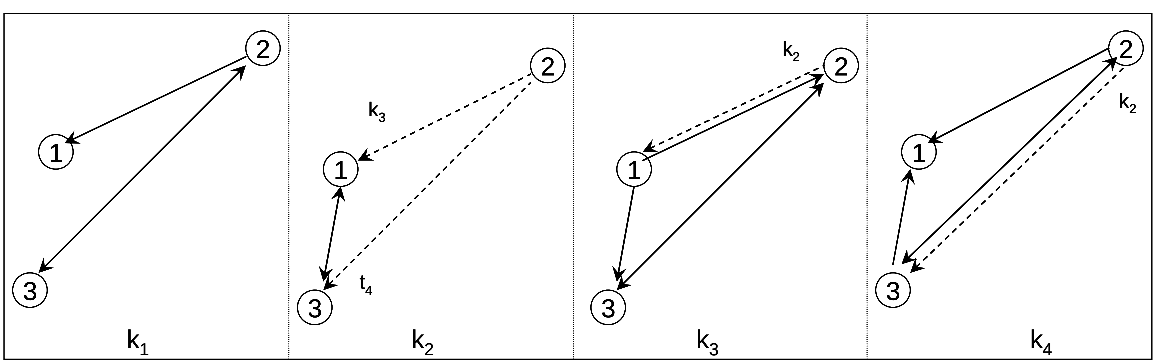

Figure 4: Example Communications Events across Four Time Steps.

Converting the communication events of the four time steps into an Information Graph produces Figure 5. From this view, several features can be seen. First, successful communication events occur in purely random patterns for this example. Second, some messages are delayed across several time steps, specifically from (2,k_2) to (1,k_3) and (3,k_4). Finally, some messages are delivered out of order due to time-varying delays between agents 2 and 3, where the message from (2,k_3) arrives before the message from (2,k_2) at agent 3.

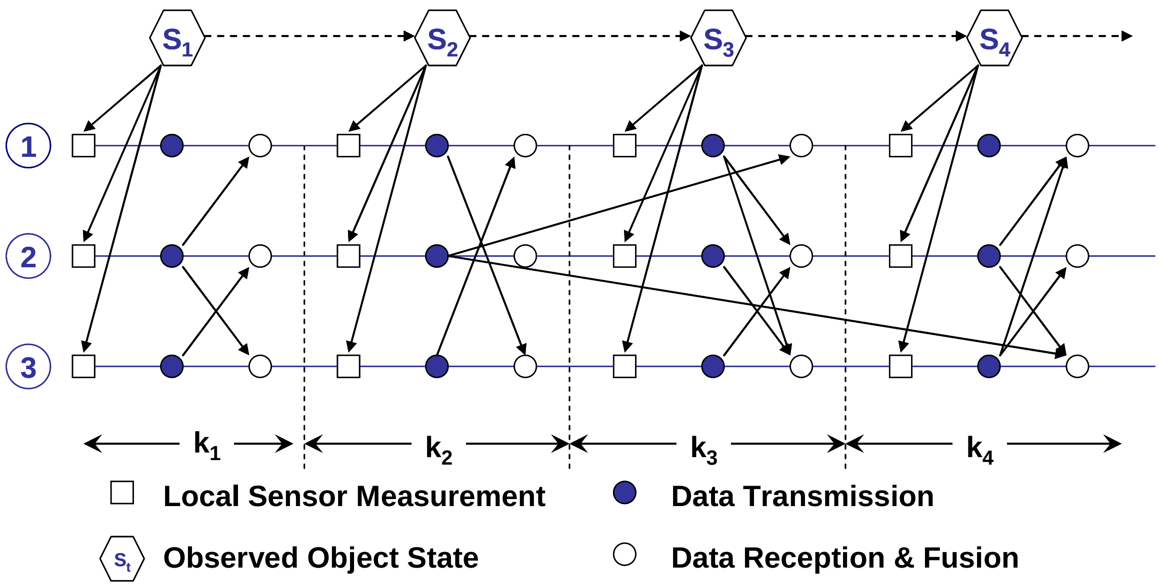

Figure 5: Example Information Graph across Four Time Steps.

Before beginning the derivation, an explanation of the notational conventions and simplifying assumptions used throughout this process is in order. While there are many similarities to those used in [36,37] for the Information Graph, some modifications and clarifications are made to keep the Local Fusion Graph representation compact. While all the nodes of an information graph can be represented in the local fusion graphs, only a single node representing each local estimate is used. It will be demonstrated that agents only need to exchange local estimates and their interrelationships to maintain distributed fusion capabilities without loss of information. As discussed in [39], measurements may be conditionally dependent due to state variables. In such cases, the fusion graphs should contain measurement nodes as part of the representation. To maintain clarity in the derivation, however, measurements nodes will be excluded at this juncture but will be addressed later.

The key assumption of the approach developed here is that data exchanges between agents include sets of data nodes and pointers. The nodes contain the data of interest (e.g., local estimates) generated locally by a fusion agent, and the pointers connect the nodes to form a network, thus indicating the pedigree of the data. In wireless networks, the size of data communications is a large concern, so transmitting the minimum data set is desired. The factors influencing communications protocols and their impacts on communications requirements are presented in Section 3.3. For the purposes of the derivation, full data sets will be used to demonstrate how they are combined.

To begin the derivation, a network representation of the communications (or fusion) events at time k_1 is created for each of the fusion agents based on Figure 4. Each agent fuses the incoming data with its local information and stores the data in memory along with data indicating the sending agent and time references as shown in Figure 6. Note that the subscript "f" is used in the figures to indicate fused data and is only a temporary node in the graphs, while the remaining nodes depict local data.

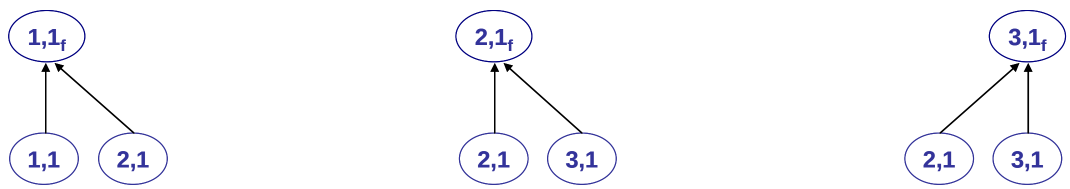

Figure 6: Local Fusion Graphs for Time k_1.

At time k_2, each agent uses the new fused information from k_1 to calculate the local data based on time updates and measurements. The agents then exchange data in accordance with the events shown in Figure 4. At this time step, agent 2 sends data to agents 1 and 3, but the data delivery is delayed beyond the current time step. Agent 2 does not receive any incoming data and thus performs no fusion operations at time k_2. Agent 1, however, receives data from agent 3 and fuses it with its local information. Before the data can be fused, agent 1 must identify any shared information contained in the data sets. As shown in Figure 7, the two sets both contain data for $(2,k_1)$, which is the data associated with node 2 at time k_1. The time-updated value of this data must be removed as in Equation (1) to create a complete and non-duplicative data set:

$$\begin{aligned}\Phi\left(I_1(t_2)\right) &= \Phi\left(I_1(t_2 \mid t_2) \cup I_3(t_2 \mid t_2)\right) \\ &= \Phi\left(I_1(t_2 \mid t_2)\right) + \Phi\left(I_3(t_2 \mid t_2)\right) - \Phi\left(\bar{I}_{1,3}(t_2)\right) \\ &= \Phi\left(I_1(t_2 \mid t_2)\right) + \Phi\left(I_3(t_2 \mid t_2)\right) - \Phi\left(I_2(t_1 \mid t_1)\right)_{t_2}\end{aligned} \tag{12}$$

Agent 1 then adds the new information to its local fusion network, providing the agent with an understanding of its data pedigree for use in future fusion events. Since agent 3 receives data from agent 1 at this time step, the above process is repeated at agent 3, resulting in the final fusion nets at time k_2 for each agent as shown in Figure 8.

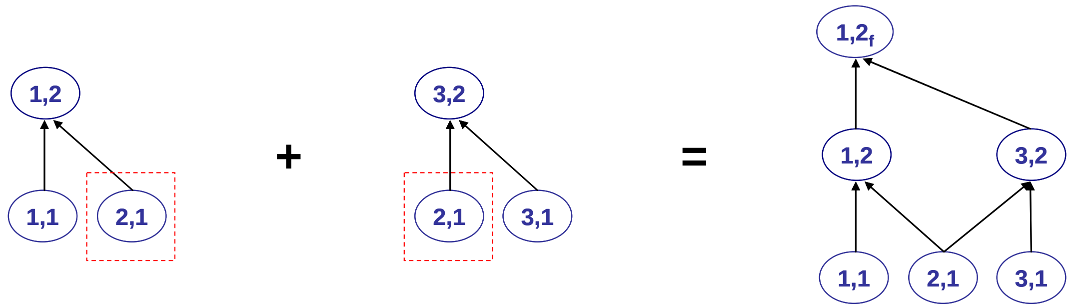

Figure 7: Fusion Event for Agent 1 at Time k_2.

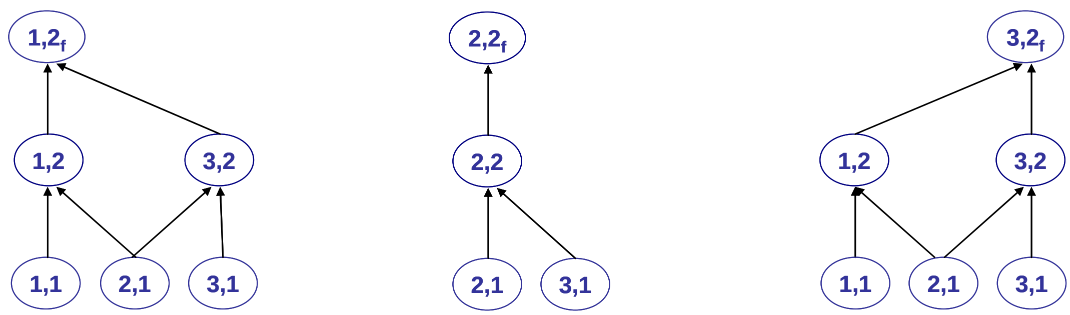

Figure 8: Local Fusion Graphs for Time k_2.

At time k_3, agent 1 receives the delayed data set associated with $(2,k_2)$. As shown in Figure 9, the common information in the two data sets is $\{(2,k_1),(3,k_1)\}$. In addition to performing a time-update operation on the common information $(2,k_1)\cup(3,k_1)$, the agent must also perform the extra step of time-updating the delayed data $(2,k_2)$ from k_2 to k_3 before in can be fused with $(1,k_3)$, resulting in the following fusion equation:

$$\Phi(I_1(k_3)) = \Phi(I_1(k_3 \mid k_3)) + \Phi(I_2(k_2 \mid k_2))_{k_3} - [\Phi(I_2(k_1 \mid k_1)) + \Phi(I_3(k_1 \mid k_1))]_{k_3} \tag{13}$$

And since it has been shown in the previous time step and from Figure 9 that:

$$\Phi(I_2(k_1 \mid k_1)) + \Phi(I_3(k_1 \mid k_1)) = \Phi(I_2(k_2)) \tag{14}$$

Equation (13) becomes:

$$\Phi(I_1(k_3)) = \Phi(I_1(k_3 \mid k_3)) + \Phi(I_2(k_2 \mid k_2))_{k_3} - \Phi(I_2(k_2))_{k_3} \tag{15}$$

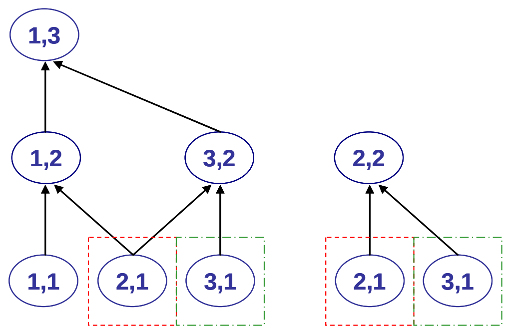

Figure 9: Common Information for $(1,k_3) \cup (2,k_2)$.

Agent 2 receives data from agents 1 and 3 at k_3. By the set theory described in Equation (1), common data must be identified in all possible combinations of data sets: $\{(1,k_3),(2,k_3)\}$, $\{(1,k_3),(3,k_3)\}$, $\{(2,k_3),(3,k_3)\}$, and $\{(1,k_3),(2,k_3),(3,k_3)\}$. The common information sets from the three pairwise combinations are shown in Figure 10. The combinations $(1,k_3) \cup (2,k_3)$ and $(2,k_3) \cup (3,k_3)$ both produce the same common sets. The combination $(1,k_3) \cup (3,k_3)$, however, has a more complex solution because the common sets $\{(1,k_2),(3,k_2)\}$ have prior information that must also be searched. This secondary search yields $(2,k_1)$ as shown in the bottom of Figure 10. In Figure 11, the joint common information across all three data sets is shown to be $\{(2,k_1),(3,k_1)\}$. The resulting fusion equation is given as:

$$\Phi\left(I_2(k_3)\right) = \sum_{i=1}^{3}\Phi\left(I_i(k_3 \mid k_3)\right) - \left[\Phi\left(\bar{I}_{1,2}(k_3)\right) + \Phi\left(\bar{I}_{1,3}(k_3)\right) + \Phi\left(\bar{I}_{2,3}(k_3)\right)\right]_{k_3} + \Phi\left(\bar{I}_{1,2,3}(k_3)\right)_{k_3} \tag{16}$$

where:

$$\Phi\left(\bar{I}_{1,2}(k_3)\right) = \Phi\left(I_2(k_1 \mid k_1)\right) + \Phi\left(I_3(k_1 \mid k_1)\right)$$
$$\Phi\left(\bar{I}_{1,3}(k_3)\right) = \Phi\left(I_1(k_2 \mid k_2)\right) + \Phi\left(I_3(k_2 \mid k_2)\right) - \Phi\left(I_2(k_1 \mid k_1)\right)_{k_2}$$
$$\Phi\left(\bar{I}_{2,3}(k_3)\right) = \Phi\left(I_2(k_1 \mid k_1)\right) + \Phi\left(I_3(k_1 \mid k_1)\right)$$
$$\Phi\left(\bar{I}_{1,2,3}(k_3)\right) = \Phi\left(I_2(k_1 \mid k_1)\right) + \Phi\left(I_3(k_1 \mid k_1)\right)$$

Since identical data sets are fused at agents 2 and 3, $\Phi\left(I_3(k_3)\right)$ is given as Equation (16).

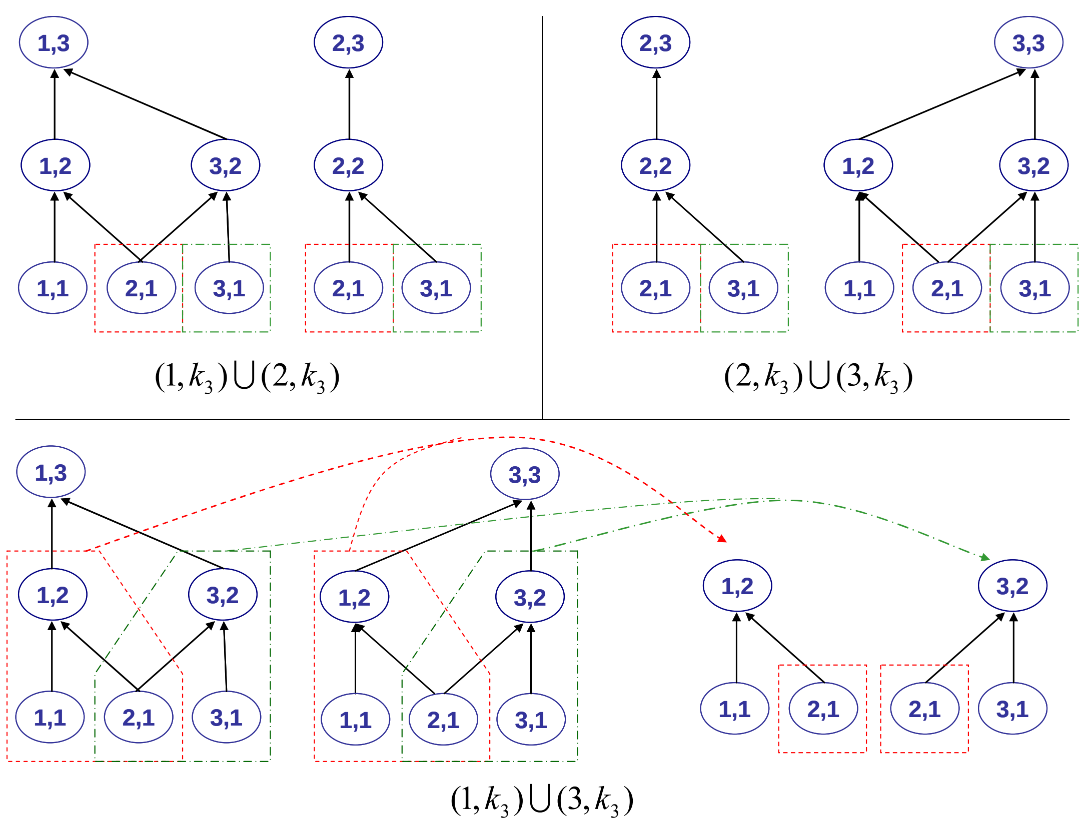

Figure 10: Common Information for $(1,k_3)\cup(2,k_3)$, $(2,k_3)\cup(3,k_3)$, and $(1,k_3)\cup(3,k_3)$.

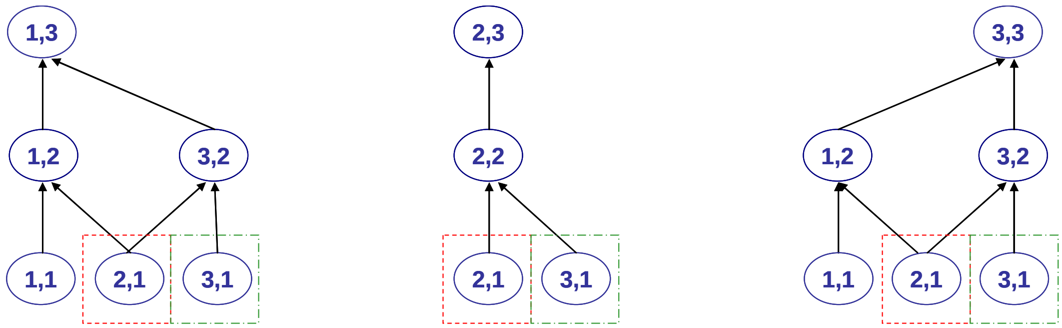

Figure 11: Common Information for $(1,k_3) \cup (2,k_3) \cup (3,k_3)$.

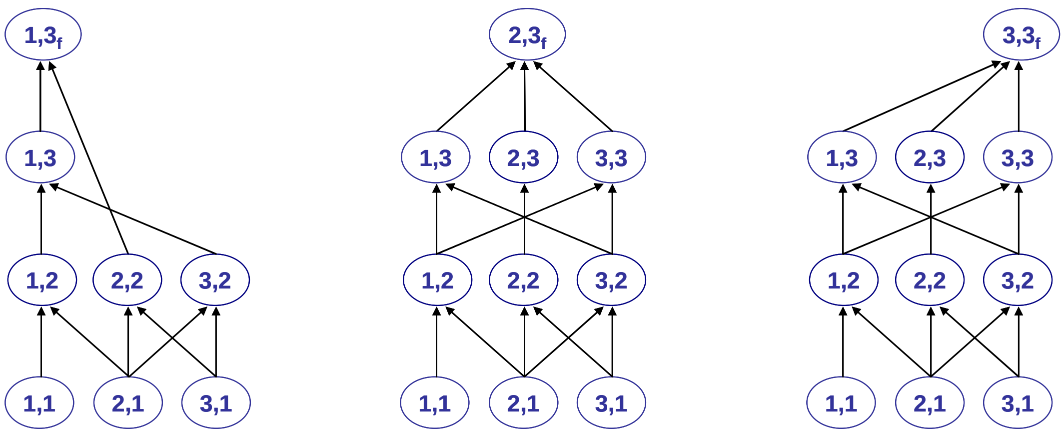

Figure 12: Local Fusion Graphs for Time k_3.

For time k_4, agent 1 receives data associated with $(2,k_4)$ and $(3,k_4)$. The shared data among $(1,k_4)$ and $(2,k_4)$ is shown in Figure 13 to be $\{(1,k_3),(2,k_2)\}$ with the subsequent common information $(1,k_3) \cap (2,k_2)$ shown to be $\{(2,k_1),(3,k_1)\}$. Inspection of the history associated with $(3,k_4)$ shows a history identical to $(2,k_4)$. Thus, the same results are found for $(1,k_4) \cap (3,k_4)$ as with $(1,k_4) \cap (2,k_4)$. Additionally, the shared information $(2,k_4) \cap (3,k_4)$ is $\{(1,k_3),(2,k_3),(3,k_3)\}$, which is equivalent to the fused results for agents 2 and 3 at k_3. Referring again to Figure 13, the common information shared among all three data sets to be fused is found to be $\{(1,k_3),(2,k_2)\}$ with $\{(2,k_1),(3,k_1)\}$ as the subsequent common information $(1,k_3)\cap(2,k_2)$. Thus the resulting equation for agent 1 is given as:

$$\begin{aligned}\Phi\left(I_1(k_4)\right) = \sum_{i=1}^{3}\Phi\left(I_i(k_4\mid k_4)\right) - \left[\Phi\left(\bar{I}_{1,2}(k_4)\right)+\Phi\left(\bar{I}_{1,3}(k_4)\right)+\Phi\left(\bar{I}_{2,3}(k_4)\right)\right]_{k_4} \\ +\Phi\left(\bar{I}_{1,2,3}(k_4)\right)_{k_4}\end{aligned} \tag{17}$$

where:

$$\begin{aligned}
\Phi\left(\bar{I}_{1,2}(k_4)\right) &= \Phi\left(I_1(k_3\mid k_3)\right)+\Phi\left(I_2(k_2\mid k_2)\right)_{k_3} - \left[\Phi\left(I_2(k_1\mid k_1)\right)+\Phi\left(I_3(k_1\mid k_1)\right)\right]_{k_3} \\
\Phi\left(\bar{I}_{1,3}(k_4)\right) &= \Phi\left(I_1(k_3\mid k_3)\right)+\Phi\left(I_2(k_2\mid k_2)\right)_{k_3} - \left[\Phi\left(I_2(k_1\mid k_1)\right)+\Phi\left(I_3(k_1\mid k_1)\right)\right]_{k_3} \\
\Phi\left(\bar{I}_{2,3}(k_4)\right) &= \Phi\left(I_2(k_3)\right)_{t_4} = \Phi\left(I_2(k_4\mid k_3)\right) \quad or \quad \Phi\left(I_3(k_3)\right)_{t_4} = \Phi\left(I_3(k_4\mid k_3)\right) \\
\Phi\left(\bar{I}_{1,2,3}(k_4)\right) &= \Phi\left(I_1(k_3\mid k_3)\right)+\Phi\left(I_2(k_2\mid k_2)\right)_{k_3} - \left[\Phi\left(I_2(k_1\mid k_1)\right)+\Phi\left(I_3(k_1\mid k_1)\right)\right]_{k_3}
\end{aligned}$$

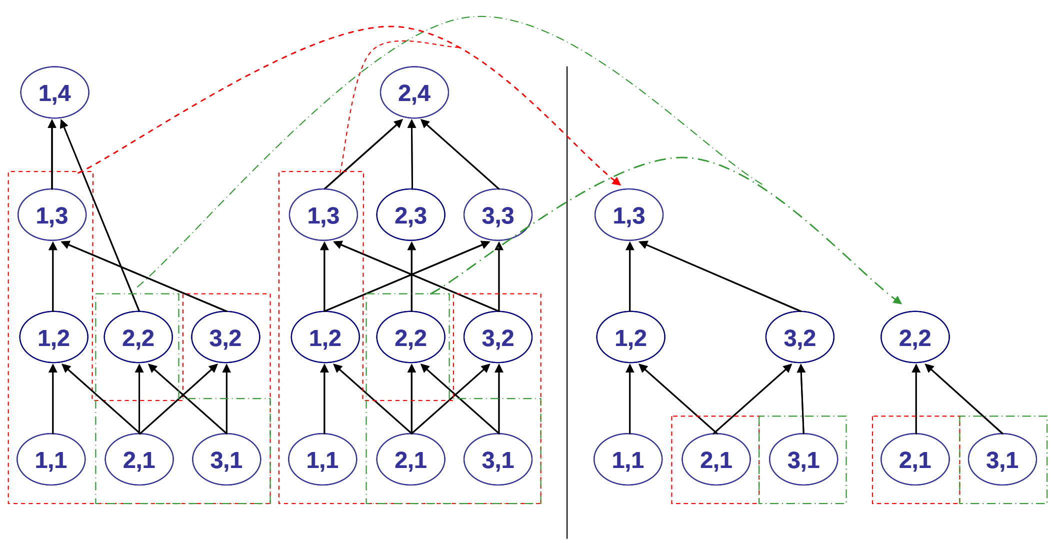

Figure 13: Common Information for $(1,k_4)\cup(2,k_4)$.

Agent 2 receives data only from agent 3 at k_4. From the prior analysis for agent 1, it is easily shown that:

$$\begin{aligned}\Phi(I_2(k_4)) &= \Phi(I_2(k_4 \mid k_4)) + \Phi(I_3(k_4 \mid k_4)) - \Phi(\bar{I}_{2,3}(k_4)) \\ &= \Phi(I_2(k_4 \mid k_4)) + \Phi(I_3(k_4 \mid k_4)) - \Phi(I_2(k_4 \mid k_3))\end{aligned} \tag{18}$$

Agent 3 receives data from agent 2 at the current time step, but Figure 4 and Figure 5 show that it also receives the delayed data set (2,k_2). Figure 14 provides an analysis of the common information sets for this scenario. It is shown that (2,k_2) is common among all the data sets, and (2,k_4)$\cap$(3,k_4) has been previously found to be equivalent the fused results for agents 2 and 3 at k_3. Representing the delayed data with the subscript "d" the fusion equation that results is then given as:

$$\begin{aligned}\Phi(I_3(k_4)) = {} & \Phi(I_2(k_4 \mid k_4)) + \Phi(I_3(k_4 \mid k_4)) + \Phi(I_2(k_2 \mid k_2))_{k_4} - \\ & \left(\Phi(\bar{I}_{2,3}(k_4)) + \Phi(\bar{I}_{2,2_d}(k_4)) + \Phi(\bar{I}_{2_d,3}(k_4))\right) + \Phi(\bar{I}_{2,2_d,3}(k_4))\end{aligned} \tag{19}$$

where:

$$\begin{aligned}&\Phi(\bar{I}_{2,2_d}(k_4)) = \Phi(I_2(k_2 \mid k_2))_{k_4} \\ &\Phi(\bar{I}_{2,3}(k_4)) = \Phi(I_2(k_3))_{k_4} = \Phi(I_2(k_4 \mid k_3)) \quad or \quad \Phi(I_3(k_3))_{k_4} = \Phi(I_3(k_4 \mid k_3)) \\ &\Phi(\bar{I}_{2_d,3}(k_4)) = \Phi(I_2(k_2 \mid k_2))_{k_4} \\ &\Phi(\bar{I}_{2,2_d,3}(k_4)) = \Phi(I_2(k_2 \mid k_2))_{k_4}\end{aligned}$$

Combining these results yields:

$$\Phi(I_3(k_4)) = \Phi(I_2(k_4 \mid k_4)) + \Phi(I_3(k_4 \mid k_4)) - \Phi(I_3(k_4 \mid k_3)) \tag{20}$$

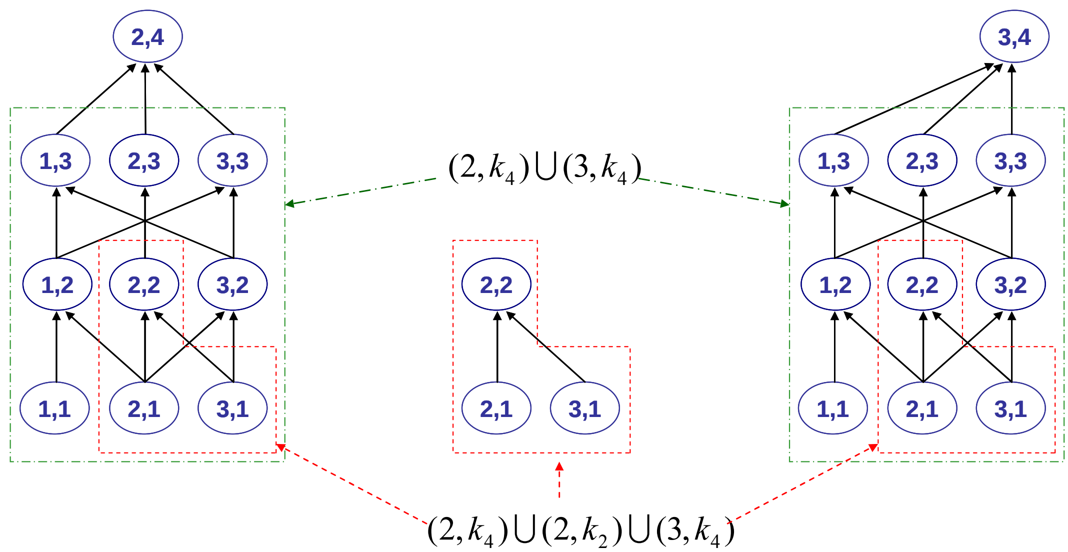

Figure 14: Common Information for $(2,k_4)\cup(3,k_4)$ and $(2,k_4)\cup(2,k_2)\cup(3,k_4)$.

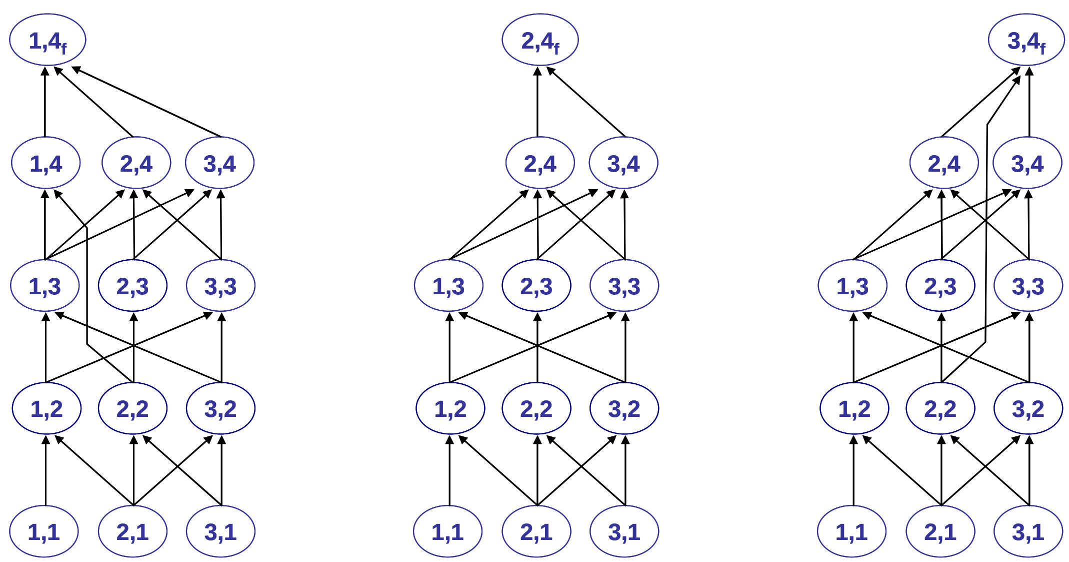

Figure 15: Local Fusion Graphs for Time k_4.

The process described above for developing the local fusion graphs and using them to formulate the fusion equations can be summarized in the following steps [49]:

1) Select a node from the receiving agent's local graph starting at the most recent time and search the received fusion data to determine if common information exists.
2) If common information is found:
   - Store the common information for fusion calculations per Equations (2) and (3).
   - Prune any common information (nodes and pointers) from the received data.
   - Remove the ancestral information of the common node from further use in the current selection process.
3) Repeat the selection and search (steps 1 and 2) to find all common elements of the two fusion graphs.
4) Fuse the new information in the incoming data set elements into agent's local fusion graph.
5) If two or more common nodes were found in the original search, then repeat steps 1-3 to detect subsequent common elements for each of the node combinations per Equation (1). Recursively continue steps 1-3 until the search terminates at a single node or no further common nodes are found.
6) Apply the sequencing of the common information to Equations (2) and (3) to produce the fusion equations.

The Local Fusion Graph algorithm preserves the data and their relationships with each other as required for distributed data fusion. As previously mentioned, the example used for the derivation in does not represent measurement dependencies resulting from the observed object's state in the local fusion graphs. To maintain clarity in the derivation, no measurements dependencies were assumed. As discussed in [39], however, dependencies may exist within the fused estimates due to state model representation. These dependencies are easily incorporated into the Local Fusion Graph as shown in Figure 16.

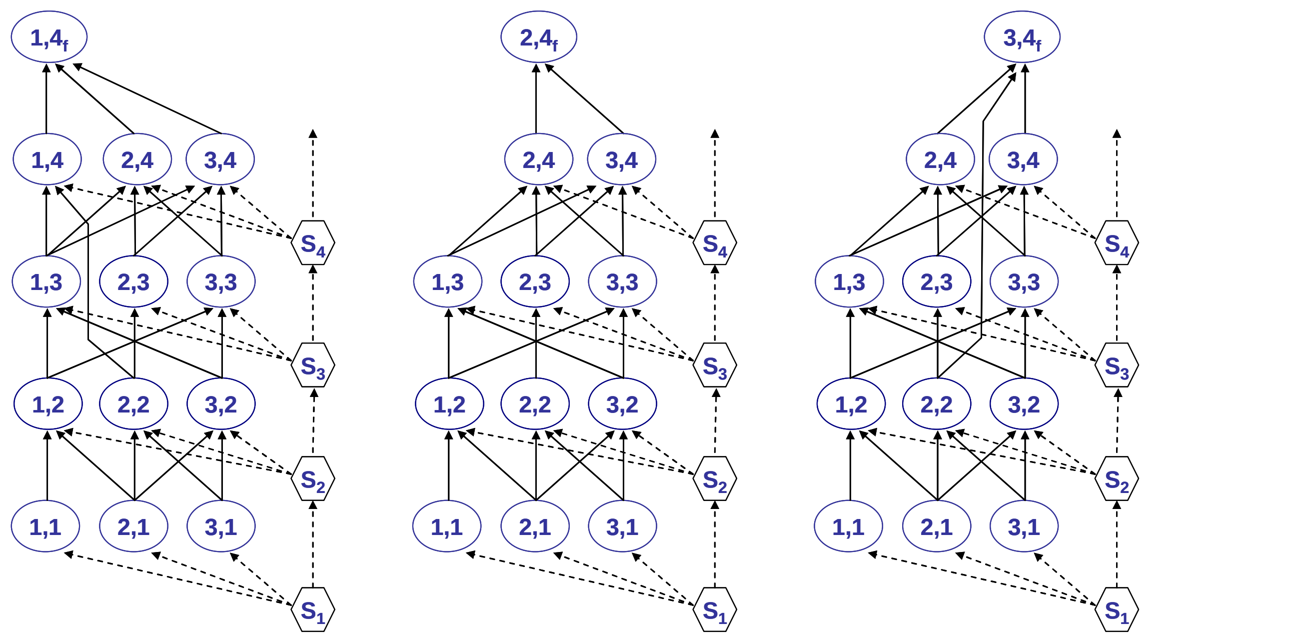

Figure 16: Example Local Fusion Graphs with Object State Dependencies.

From the figure and prior derivation of fusion equations, it is apparent that any inter-agent dependencies in the state measurements must be addressed. Given the Information Graph or Local Fusion Graph methods, these dependencies can potentially be identified and removed a priori per [39]. Another technique for dependency identification in linear estimation systems using Kalman Filter (KF) techniques is by deriving the virtual measurement using the technique described in [46]. Thus each agent would need to remove the measurements dependencies prior to fusing the estimates using the operations presented in the previous section. While the mathematical operations differ between removal of dependencies due to measurements and dependencies due to communications, both types of dependencies are found using the same technique for the Local Fusion Graph.

2.5 Local Fusion Graph Summary

As discussed in the introduction, distributed information fusion implementations must address the realistic characteristics of wireless networks. MANETS add complexity to fusion solutions due to the absence of an a priori communications structure. The non-deterministic nature of these networks results in uncertain message delivery and out-of-sequence data. Distributed fusion techniques found to date, however, are not generally applicable to these dynamic communications architectures.

The Local Fusion Graph approach is proposed here as a general solution to distributed fusion. It makes no a priori assumptions regarding communications architectures, network membership, or network connectivity. Its ability to handle arbitrary network connectivity and message delays without a prior knowledge allows fusion agents to operate collaboratively as well as independently. Its foundations in the logic of the Information Graph approach [36,37] lend to the pedigree of the technique.

An analysis of estimation performance using the Local Fusion Graph algorithm is presented in the next chapter. The algorithms are implemented in a computer-based model, and results are analyzed to show consistency with other techniques where possible. Insights into estimation performance in networks with ad hoc connectivity and stochastic link characteristics are also discussed along with communications requirements for the Local Fusion Graph.

3 SIMULATION OF THE LOCAL FUSION GRAPH METHOD AND ANALYSES OF RESULTS

The Local Fusion Graph approach was developed into a computer-based model to validate and assess the methodology's performance and to understand the basic characteristics of estimation capabilities in distributed fusion networks with varying degrees of average connectivity. A first-order assessment of data exchange needs is also made to understand the communications impacts of the method.

The performance comparisons are made relative to known solutions in order to validate the results of the Local Fusion Graph method. Because no general solutions are available for uncertain network connectivity, broadcast and cyclical communications architectures are used for comparison with the Local Fusion Graph under full connectivity conditions. Similarly, stand-alone estimation solutions are used for comparison under conditions with no connectivity.

Distributed fusion characteristics in stochastic networking conditions are explored through investigations of estimation capabilities with respect to time and network connectivity. In addition to the broadcast and cyclical architectures that are used for validation purposes, an ad hoc architecture is used for estimation performance examinations. The analyses focus on ascertaining general trends in estimation performance as evidenced by state estimates, average estimation errors, and estimation covariance. The analyses are conducted under transient and steady-state estimation conditions.

Because data communications are a major factor in distributed fusion operations, the communications requirements of the Local Fusion Graph are explored. The assessments provide insight into the relative impact of network connectivity on data exchange requirements among the fusion agents. The resulting relationship between data exchange and estimation performance is also explored. For comparison purposes, the communications requirements and resulting estimation performance of the Local Fusion Graph are compared with those of the naïve fusion algorithm.

The simulation model used for the analysis is described in the next section. The subsequent two sections provide an analysis of the estimation results and communications requirements.

3.1 Model Description

The Local Fusion Graph algorithm, known solutions, and KF estimation formulations were developed in the MATLAB® programming environment. The resulting routines can be applied to any distributed fusion network regardless of communications patterns and number of fusion agents. To complete the model, a communications event generator was created to feed the fusion routine and the resulting output is applied to a linear KF estimation routine. The model uses a Monte Carlo approach to randomize communications events over a user-defined number of simulation runs and determine average outcomes. A functional block diagram of the resulting system is shown in Figure 17.

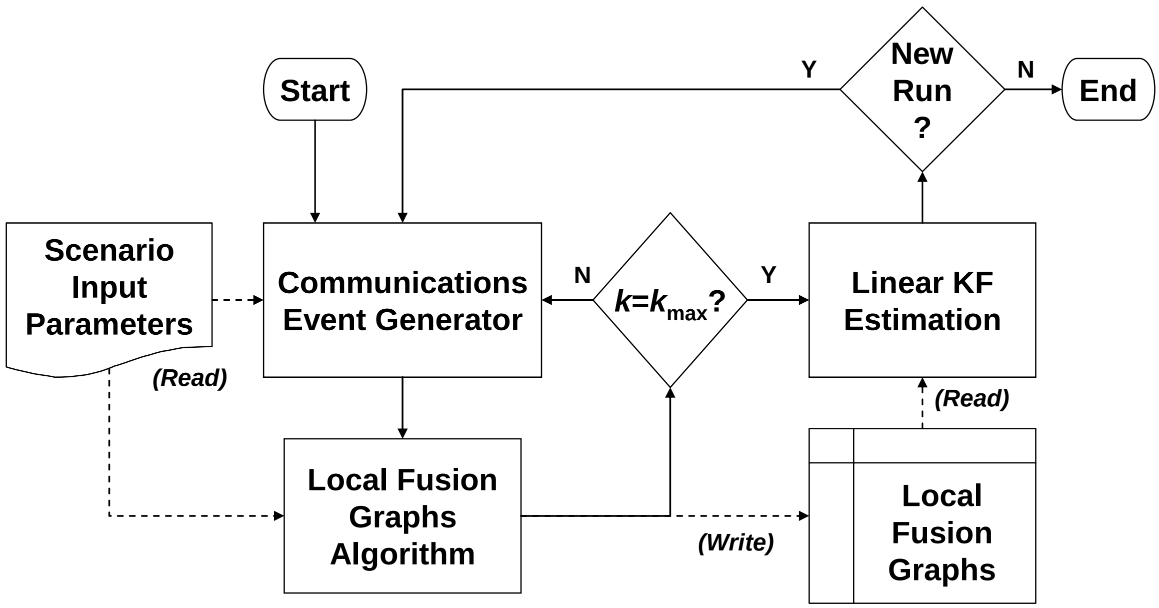

Figure 17: Local Fusion Graph Simulation Model Block Diagram.

The scenario input parameters define the particular options to be used in the scenario. The scenario parameters include the models and associated parameter values that define the true state to be estimated as well as the measurements model for each time step in the scenario. For the results of this study, linear state and measurement models used by the estimation routine are the familiar form given as:

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)\mathbf{v}(k) \tag{21}$$

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) \tag{22}$$

where the notation is as follows:

$\mathbf{x}(k)$: state vector at time k

$\mathbf{F}(k)$: state transition matrix at time k

$\mathbf{B}(k)$: input control matrix at time k

$\mathbf{u}(k)$: input (or forcing) vector at time k

$\mathbf{G}(k)$: process noise gain matrix at time k

$\mathbf{v}(k)$: process noise vector at time k

$\mathbf{z}(k)$: measurements (or observation) vector at time k

$\mathbf{H}(k)$: measurements (or observation) model at time k

$\mathbf{w}(k)$: measurements noise vector at time k

For the purposes of the model, the noises are assumed to be zero mean Gaussian, that is $\mathbf{v}(k) \sim N(0, \mathbf{Q}(k))$ and $\mathbf{w}(k) \sim N(0, \mathbf{R}(k))$.

The communications event generator develops a communications connectivity matrix at each simulation time step to determine successful message deliveries between each pair of agents. The module utilizes a comparison between average connectivity statistics given for the scenario and a random number generated from a uniform distribution function. The model applies the random draws at each time step to the fusion agents according to the assumptions of the communications architecture.

As previously mentioned, three architectures are modeled. The basic ad hoc architecture modeled here is a point-to-multipoint communication system. Fusion agents in the ad hoc network transmit messages that are intended for one or more other agents in the network. The broadcast network also employs a point-to-multipoint scheme, but transmitted messages are intended for all other agents in the scenario. In the cyclic architecture, each fusion agent communicates with only one other agent throughout the scenario, thus creating a network as shown in Figure 18.

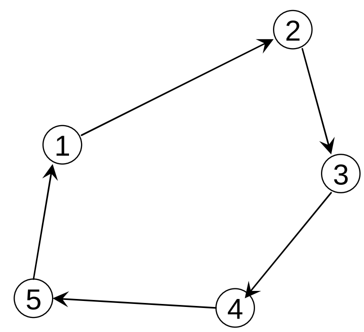

Figure 18: Cyclic Fusion Network Architecture.

The communications event generator uses the three basic architectures as templates to which it applies random connectivity effects. For the ad hoc communications architecture, a random draw is performed for each possible transmitter-receiver pair at each time step. A successful draw signifies that the message is successfully delivered from the sending agent to the receiving agent. The result models a multicast transmission scheme where some intended receivers may not receive the message due to any number of wireless communications phenomena that affect message reception.

Under the broadcast communications architecture, a random draw is performed for each transmitter at each time step. If the draw indicates a success, then the agent transmits a message that is received by all other agents in the scenario. Otherwise no transmission occurs. The effect is that of a broadcast scheme with an average transmit duty cycle and assured delivery given the transmission.

Because each fusion agent communicates with only one other the cyclic communications architecture, the random draw is applied only to the appropriate transmitter-receiver pairs. In this case, the random draw can be interpreted as either the result of propagation effects or an average transmission duty cycle.

Once the communications event matrix is completed, the fusion routine receives the communications matrix for the current time step and builds the local fusion networks for each agent. It does so by combining the each agent's locally-stored fusion network with the fusion networks received from other agents. The resulting new information is stored along with an associated vector containing the common nodes and pointers required for constructing the fusion equation.

The estimation routine is initiated after the Local Fusion Graph routine has iterated through the complete time history for each node. The estimation routine iterates through each time step, performing a time-update and measurements update using a KF algorithm:

$$\hat{\mathbf{x}}(k \mid k-1) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)\mathbf{v}(k) \tag{23}$$

$$\mathbf{P}(k \mid k-1) = \mathbf{F}(k)\mathbf{P}(k-1)\mathbf{F}'(k) + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}'(k) \tag{24}$$

and

$$\hat{\mathbf{x}}(k \mid k) = \hat{\mathbf{x}}(k \mid k-1) + \mathbf{K}(k)\big(\mathbf{z}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k \mid k-1)\big) \tag{25}$$

$$\mathbf{P}(k \mid k) = \big(\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\big)\mathbf{P}(k \mid k-1) \tag{26}$$

for:

$$\mathbf{K}(k) = \mathbf{P}(k \mid k-1)\mathbf{H}^T(k)\big[\mathbf{H}(k)\mathbf{P}(k \mid k-1)\mathbf{H}^T(k) + \mathbf{R}(k)\big]^{-1} \tag{27}$$

The estimation module then develops the fusion equations for each agent at each time step according to Equations (2) and (3) and uses the local estimates from Equations (24) and (25) as inputs. The estimation also performs time-updates on the common prior information contained in the data sets being fused. These time-updated estimates relating to common data in the fusion set are incorporated into the fusion algorithm. The results are stored for output to data files as well as a standard set of graphs.

As a means of comparison, the estimation module also generates covariance and state estimates from known formulae for the broadcast and cyclic communications architectures [36]. The results of these calculations can be compared to those produced by the Local Fusion Graph algorithms under full network connectivity to validate results. Similarly, stand-alone (single agent) predictions are generated based on a standard KF routine for validation of the Local Fusion Graph algorithms under the condition of no network connectivity. The equations used for fusion in broadcast and cyclical communications architectures are covered quite extensively in [36,37], so only the resulting formulations used in the model are presented here.

In the broadcast communications architecture, each agent sends its most recent local estimate to every other agent in the network for every time step [36,37]. For time k_1, no prior common information exists and the fusion equations simplify to the following:

$$\mathbf{P}_j(k_1) = \left[\sum_{i=1}^{n} \mathbf{P}_i^{-1}(k_1 \mid k_1) \right]^{-1} \tag{28}$$

$$\hat{\mathbf{x}}_j(k_1) = \mathbf{P}_j(k_1)\left[\sum_{i=1}^{n} \mathbf{P}_i^{-1}(k_1 \mid k_1)\hat{\mathbf{x}}_i(k_1 \mid k_1)\right] \tag{29}$$

For times greater than k_1, the common prior information is found to be the prior fused estimate, resulting in the following fusion equations [36,37]:

$$\mathbf{P}_j(k) = \left[\sum_{i=1}^{n} \mathbf{P}_i^{-1}(k \mid k) - \sum_{\substack{i=1 \\ i \neq j}}^{n} \mathbf{P}_i^{-1}(k \mid k-1)\right]^{-1} \tag{30}$$

$$\hat{\mathbf{x}}_j(k) = \mathbf{P}_j(k)\left[\sum_{i=1}^{n} \mathbf{P}_i^{-1}(k \mid k)\hat{\mathbf{x}}_i(k \mid k) - \sum_{\substack{i=1 \\ i \neq j}}^{n} \mathbf{P}_i^{-1}(k \mid k-1)\hat{\mathbf{x}}_i(k \mid k-1)\right] \tag{31}$$

In cyclic communications architectures [36,37], each fusion agent sends its local estimate to only one other agent creating a singly-connected network as shown in Figure 18. For time k_1, no priors exist and the formulation is equivalent to Equations (28) and (29). The basic equation for all other times is given by Equations (2) and (3) where the prior information terms are time updates of the local estimates per Equations (23) and (24). The prior information terms change for the first few time steps until the pattern is complete. For time k_2 the fusion equations are given as[1]:

$$\overline{\mathbf{P}}_j(k_2) = \mathbf{P}_{j+1}(k_1 \mid k_1)_{k_2} = \mathbf{F}(k_2)\mathbf{P}_{j+1}(k_1 \mid k_1)\mathbf{F}'(k_2) + \mathbf{G}(k_2)\mathbf{Q}(k_2)\mathbf{G}'(k_2) \tag{32}$$

$$\overline{\mathbf{x}}_j(k_2) = \hat{\mathbf{x}}_{j+1}(k_1 \mid k_1)_{k_2} = \mathbf{F}(k_2)\hat{\mathbf{x}}_{j+1}(k_1 \mid k_1) + \mathbf{B}(k_2)\mathbf{u}(k_2) + \mathbf{G}(k_2)\mathbf{v}(k_2) \tag{33}$$

---

[1] Note that for $j=n$, the index $j+1$ refers to the initial index $j=1$

For time k_3, the common prior covariance information is given by:

$$\begin{aligned}\overline{\mathbf{P}}_j(k_3) &= \left(\mathbf{P}_{j+1}^{-1}(k_2 \mid k_2) + \mathbf{P}_j^{-1}(k_1 \mid k_1)_{k_2}\right)_{k_3}^{-1} \\ &= \mathbf{F}(k_3)\left(\mathbf{P}_{j+1}^{-1}(k_2 \mid k_2) + \mathbf{P}_j^{-1}(k_1 \mid k_1)_{k_2}\right)^{-1}\mathbf{F}'(k_3) + \mathbf{G}(k_3)\mathbf{Q}(k_3)\mathbf{G}'(k_3)\end{aligned} \tag{34}$$

where

$$\mathbf{P}_j^{-1}(k_1 \mid k_1)_{k_2} = \left(\mathbf{F}(k_2)\mathbf{P}_j(k_1 \mid k_1)\mathbf{F}'(k_2) + \mathbf{G}(k_2)\mathbf{Q}(k_2)\mathbf{G}'(k_2)\right)^{-1}$$

The common prior state estimate is given by:

$$\begin{aligned}\overline{\mathbf{x}}_j(k_3) &= \overline{\mathbf{P}}_j(k_3)\left(\mathbf{P}_{j+1}^{-1}(k_2 \mid k_2)\hat{\mathbf{x}}_{j+1}(k_2 \mid k_2) + \mathbf{P}_j^{-1}(k_1 \mid k_1)_{k_2}\hat{\mathbf{x}}_j(k_1 \mid k_1)_{k_2}\right)_{k_3} \\ &= \mathbf{F}(k_3)\overline{\mathbf{P}}_j(k_3)\left(\mathbf{P}_{j+1}^{-1}(k_2 \mid k_2)\hat{\mathbf{x}}_{j+1}(k_2 \mid k_2) + \mathbf{P}_j^{-1}(k_1 \mid k_1)_{k_2}\hat{\mathbf{x}}_j(k_1 \mid k_1)_{k_2}\right) \\ &\quad + \mathbf{B}(k_3)\mathbf{u}(k_3) + \mathbf{G}(k_3)\mathbf{v}(k_3)\end{aligned} \tag{35}$$

where

$$\hat{\mathbf{x}}_j(k_1 \mid k_1)_{k_2} = \mathbf{F}(k_2)\hat{\mathbf{x}}_j(k_1 \mid k_1) + \mathbf{B}(k_2)\mathbf{u}(k_2) + \mathbf{G}(k_2)\mathbf{v}(k_2)$$

For times greater than k_4, the common prior covariance information is given by:

$$\begin{aligned}\overline{\mathbf{P}}_j(k) &= \left(\mathbf{P}_{j+1}^{-1}(k-1 \mid k-1) + \mathbf{P}_j^{-1}(k-2 \mid k-2)_{k-1} - \mathbf{P}_j^{-1}(k-3)_{k-1}\right)_k^{-1} \\ &= \mathbf{F}(k)\left(\mathbf{P}_{j+1}^{-1}(k-1 \mid k-1) + \mathbf{P}_j^{-1}(k-2 \mid k-2)_{k-1} - \mathbf{P}_j^{-1}(k-3)_{k-1}\right)^{-1}\mathbf{F}'(k) \\ &\quad + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}'(k)\end{aligned} \tag{36}$$

where

$$\mathbf{P}_j^{-1}(k-2\,|\,k-2)_{k-1} = \left[\mathbf{F}(k-1)\mathbf{P}_j(k-2\,|\,k-2)\mathbf{F}'(k-1) + \mathbf{G}(k-1)\mathbf{Q}(k-1)\mathbf{G}'(k-1)\right]^{-1}$$
$$\mathbf{P}_j^{-1}(k-3)_{k-1} = \begin{Bmatrix} \mathbf{F}(k-1)\left[\mathbf{F}(k-2)\mathbf{P}_j(k-3)\mathbf{F}'(k-2) + \mathbf{G}(k-2)\mathbf{Q}(k-2)\mathbf{G}'(k-2)\right]\mathbf{F}'(k-1) \\ +\mathbf{G}(k-1)\mathbf{Q}(k-1)\mathbf{G}'(k-1) \end{Bmatrix}^{-1}$$

The common prior state estimate is given by:

$$\overline{\mathbf{x}}_j(k) = \left\{ \mathbf{P}_j(k) \begin{pmatrix} \mathbf{P}_{j+1}^{-1}(k-1\,|\,k-1)\hat{\mathbf{x}}_{j+1}(k-1\,|\,k-1) + \\ \mathbf{P}_j^{-1}(k-2\,|\,k-2)_{k-1}\hat{\mathbf{x}}_j(k-2\,|\,k-2)_{k-1} - \\ \mathbf{P}_j^{-1}(k-3)_{k-1}\hat{\mathbf{x}}_j(k-3)_{k-1} \end{pmatrix}_k \right\} \tag{37}$$

Where the state estimates are updated to the most recent common time k-1 by:

$$\begin{aligned} \hat{\mathbf{x}}_j(k-2\,|\,k-2)_{k-1} &= \mathbf{F}(k-1)\hat{\mathbf{x}}_j(k-2\,|\,k-2) + \mathbf{B}(k-1)\mathbf{u}(k-1) + \mathbf{G}(k-1)\mathbf{v}(k-1) \\ \hat{\mathbf{x}}_j(k-3)_{k-1} &= \mathbf{F}(k-1)\big(\mathbf{F}(k-2)\hat{\mathbf{x}}_j(k-3) + \mathbf{B}(k-2)\mathbf{u}(k-2) + \mathbf{G}(k-2)\mathbf{v}(k-2)\big) \\ &\quad + \mathbf{B}(k-1)\mathbf{u}(k-1) + \mathbf{G}(k-1)\mathbf{v}(k-1) \end{aligned} \tag{38}$$

The equations for these three architectures as well as the state model form the basis of the estimation routines.

To determine averaged results, multiple simulations of the scenario are conducted in a Monte Carlo fashion. For each Monte Carlo, the randomized communications events create new communications event matrices. Thus each Monte Carlo produces new Local Fusion Graphs and estimation results. The results from all Monte Carlo simulations are averaged for graphing and analyses. Note that the model utilizes the exact same true state and measurements results in its calculations for each Monte Carlo. The state and measurements are held constant so that randomized communications events are the only factor influencing results between simulation runs. Thus the impacts of randomized communications events are easier to discern than if they were coupled with other randomized factors. Using the same true state and measurements also benefits comparisons made with an approximation method presented later in Chapter 4. The following section presents the averaged results produced by Monte Carlo simulations of the model under various conditions.

3.2 Estimation Performance Capabilities for Non-Deterministic Network Connectivity

The model and assumptions described in the previous section were exercised using the scenarios defined by the parameters presented in Table 2. The parameter values are selected to establish a simple model and allow the behavior of estimation for non-deterministic connectivity to be determined with minimal dependence on the effects of scenario complexity.

Table 2: Parameter Values for Local Fusion Graph Simulations.

Parameter	Value
B,F,G,H	1
u	0
v, Q	N(0,**Q**), 25
w, R	N(0,**R**), 100
$\mathbf{P}_{trans}(k_0)$, $\hat{\mathbf{x}}_{trans}(k_0)$	**R**, N(0,**R**)
$\mathbf{P}_{ss}(k_0)$, $\hat{\mathbf{x}}_{ss}(k_0)$	45, **x**(0)+N(0,45)

Parameter	Value
x(0)	1000+N(0,**Q**)
Monte Carlos	1000
Fusion Agents	3
k_0, k_{max}	1, 5
Message Delay	0
Network Connectivity	0, 0.25, 0.5, 0.75, 1

The simulations and analyses are separated into two scenarios, which are differentiated by the initial covariance and state estimate provided to the model. The first scenario investigates the simulation results during the transient phase of the estimation process, while the second investigates the results during steady-state estimation. The transient phase is given by the initial state and covariance parameters in Table 2 with the subscript "trans," and is defined as estimation operations that occur while estimates are well outside of the achievable limits. Under these conditions, the estimated state and covariance typically converge rapidly toward the theoretically-achievable limits along a smooth curve. The steady-state phase is defined by ongoing estimation within the theoretically-achievable state and covariance estimation limits. Under steady-state estimation conditions, state estimates do not necessarily converge in an orderly manner, but often fluctuate within a small range as a result of small changes resulting from system and measurement noises.

Each set of simulations encompasses 1000 Monte Carlo runs, and the data presented are the averages for those results. The three communications architectures discussed previously – namely ad hoc, broadcast, and cyclical architectures – are modeled and analyzed. Each Monte Carlo simulates five time steps for the selected architecture. If run for greater than five time steps, the transient estimation scenario begins to covert into steady-state conditions. The steady-state estimation scenario thus uses five time steps to maintain consistency for comparing of the results produced under the two scenarios.

The following two sections present the results of the two scenarios. The first section presents the results obtained under transient estimation conditions. The primary elements of interest in the analyses are the characteristic trends and performance of state estimations and covariance. The state estimates are presented in terms of absolute magnitude and are further characterized by the average RMS error defined here as [51]:

$$\tilde{e}_{RMS} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(x-\hat{x}\right)^2} \tag{39}$$

The second section presents the results obtained under steady-state estimation conditions. The results are assessed in a similar manner, and comparisons are made with the results obtained under transient estimation conditions to identify trends resulting from network connectivity variations.

3.2.1 Analysis of Local Fusion Graphs under Transient Estimation Conditions

The simulation of transient estimation conditions is designed to provide an assessment of distributed fusion characteristics in networks with various degrees of connectivity while the estimates are significantly outside the bounds of achievable accuracy. Under the transition conditions modeled here, the average estimates typically converge on the actual state at significant rates in a constant and consistent manner. As the estimates become closer to achievable limits, the rate of convergence decreases and eventually reaches a steady-state.

Figure 19 presents a snapshot of covariance versus time for a single simulation run of a network operating in ad hoc mode with average network connectivity varying from 0% to 100% in 25% increments. The data demonstrates the familiar non-linear reduction in covariance at 0% and 100% connectivity, with a wide fluctuation in the characteristics of the data associated with the other three connectivity levels. The randomness of the 25%, 50%, and 75% connectivity cases is the direct result of uncertain network connectivity over time. It is noted from the graph that the estimates from these three cases are bounded by the 0% and 100% connectivity cases. That result fits with the observation that the covariance of a group of agents should be no greater than the covariance of a single node, nor should the covariance ever be less than that obtained for a fully-connected network of agents for KF-based estimation.

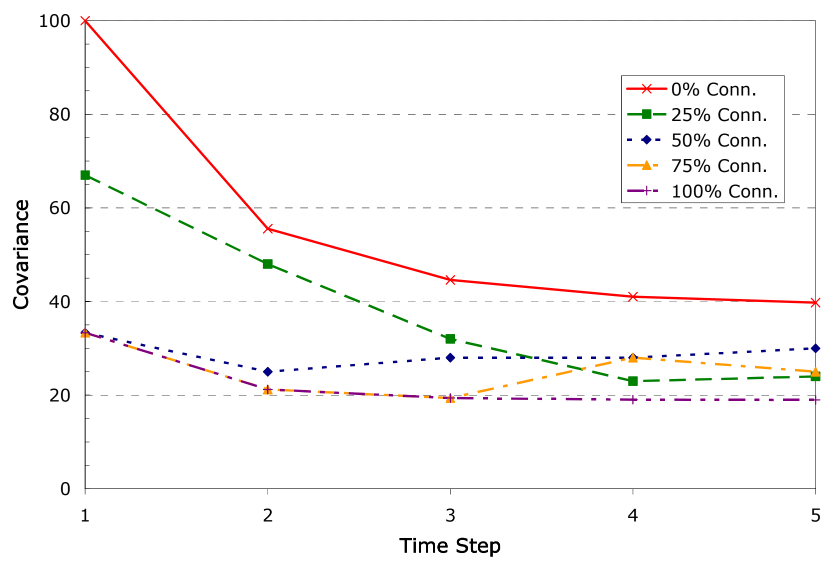

Figure 19: Sample Covariance vs. Time for Various Network Connectivity Values in an Ad Hoc Network (Single Simulation Run).

While the covariance for individual simulation runs shows arbitrary characteristics due to random network communication events, the covariance data averaged across the 1000 simulation runs as shown in Figure 20 through Figure 22 demonstrate orderly characteristics for all three network architectures. Of primary note is the absolute agreement between the event-driven results of the Local Fusion Graph method in the deterministic (i.e., 100% and 0% connectivity) cases and those of the known algorithms, thus validating the Local Fusion Graph approach. Secondly, the data sets of the three random connectivity cases demonstrate a non-linear reduction in covariance across time in all three communications architectures. The data indicate a clear dependence between the reduction in covariance and the average connectivity of the network: higher average network connectivity produces better (smaller) covariance estimates, but with performance gains that are non-linear relative to increases in average network connectivity.

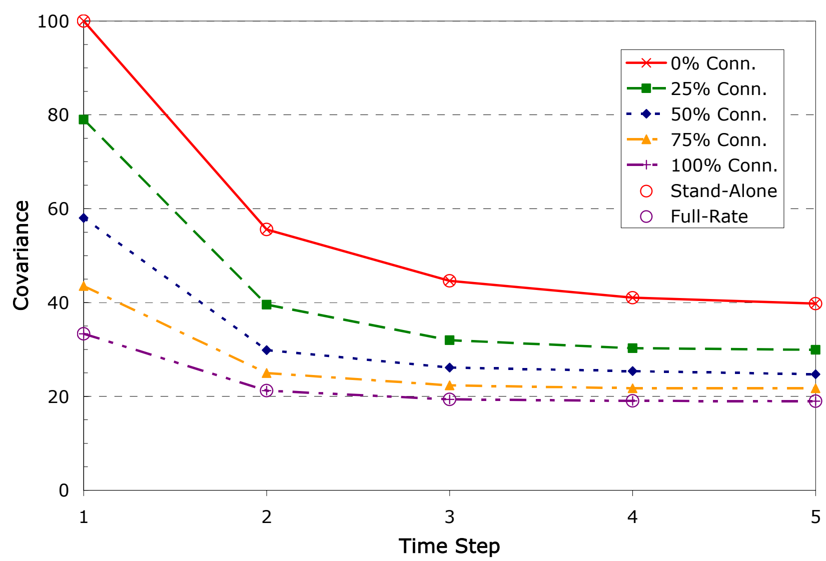

Figure 20: Covariance vs. Time for Various Network Connectivity Values in an Ad Hoc Network under Transient Estimation Conditions.

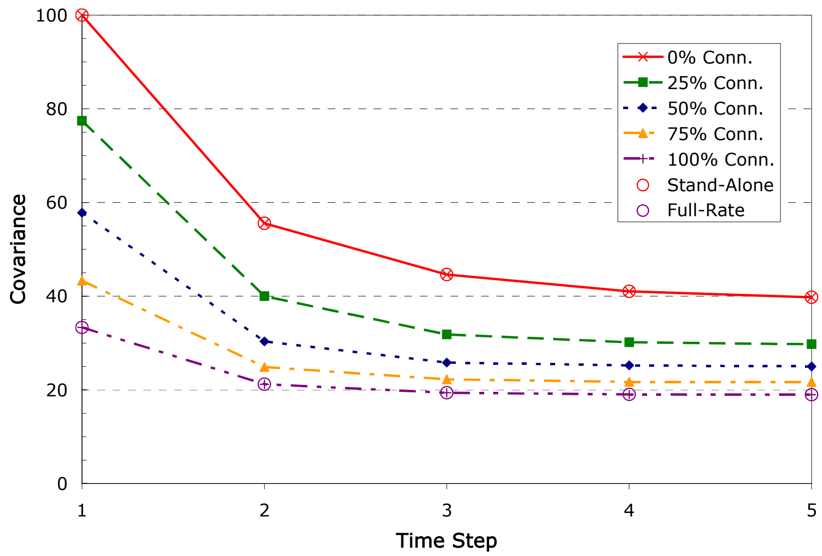

Figure 21: Covariance vs. Time for Various Network Connectivity Values in a Broadcast Network under Transient Estimation Conditions.

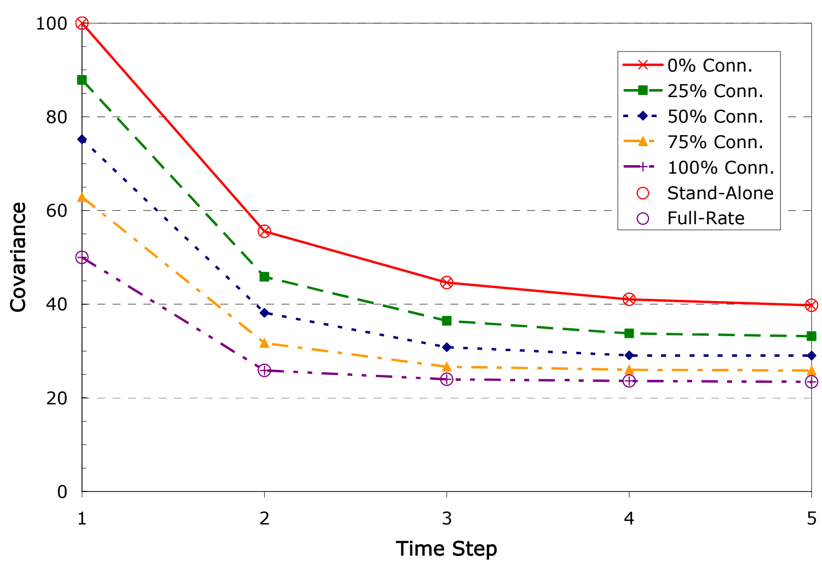

Figure 22: Covariance vs. Time for Various Network Connectivity Values in a Cyclical Network under Transient Estimation Conditions.

The data presented in Figure 23 through Figure 25 provide further insight regarding covariance as a function of network connectivity at each of the five time steps. The covariance data in these figures are normalized relative to the highest covariance (i.e., at 0% connectivity) for each associated time step to aid the comparison among time steps. The nonlinearity in the relationship between network connectivity and covariance is slight for the initial time step, and increases with time. The observed change is due to the fact that the connectivity probabilities accumulate over time. Thus the initial variances are nearly linear in proportion to network connectivity, while those at later time steps contain the compounded effect of the average network connectivity. The ad hoc and broadcast cases produce nearly identical trends, while the cyclic architecture shows a much more pronounced change between the first two time steps and a more linear relationship between connectivity and covariance at each of the time steps. In all three cases, the reduction in covariance with increasing connectivity appears to converge to a constant relationship as time progresses. The convergence is evidenced in Figure 23 through Figure 25 by the decreasing difference in the curves with each successive time step.

The resulting effect of the increasing non-linearity with time is that the degree of covariance improvement achieved with increased connectivity is slightly decreased with each time step. Thus the impact of connectivity on covariance is greatest at early time steps. A further implication is that the marginal benefit of increased network connectivity is reduced as connectivity increases and the estimate progresses toward

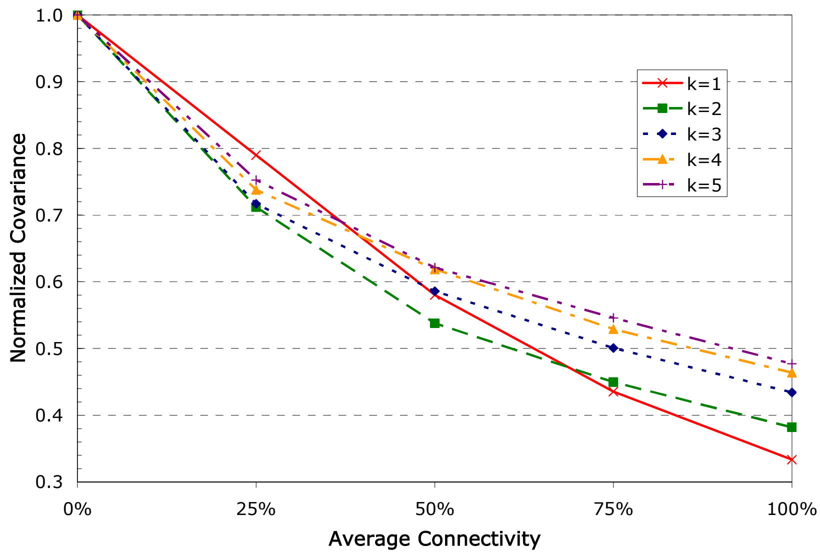

Figure 23: Normalized Covariance vs. Average Network Connectivity for Each Time Step in an Ad Hoc Network under Transient Estimation Conditions.

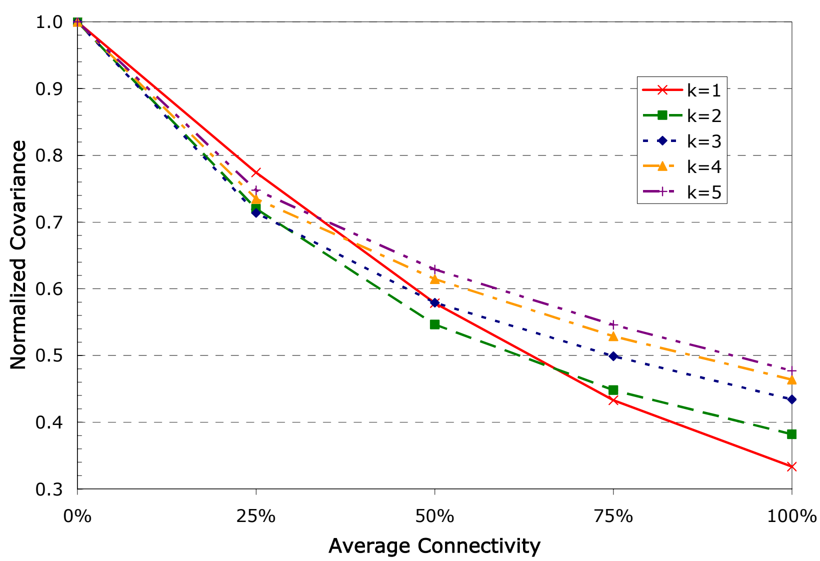

Figure 24: Normalized Covariance vs. Average Network Connectivity for Each Time Step in a Broadcast Network under Transient Estimation Conditions.

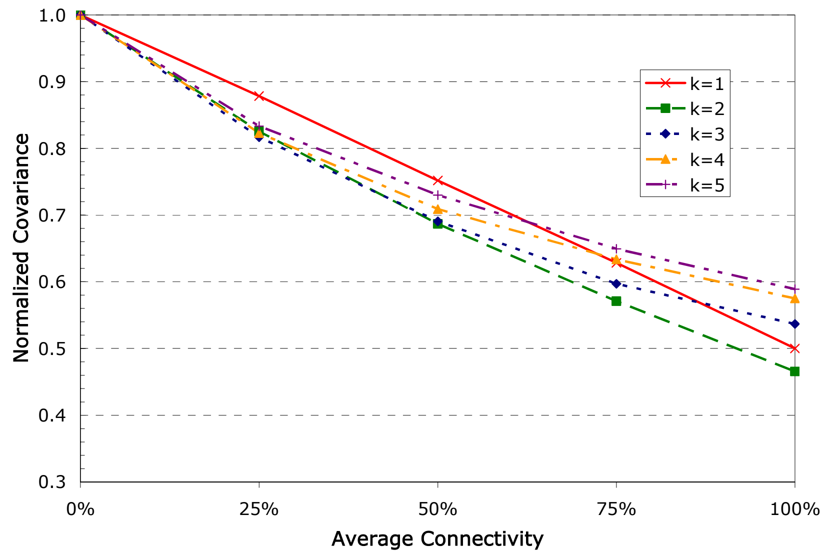

Figure 25: Normalized Covariance vs. Average Network Connectivity for Each Time Step in a Cyclical Network under Transient Estimation Conditions.

steady-state conditions. The ad hoc and broadcast data show that approximately 70% of the reduction in covariance is achieved by increasing network connectivity from 0% to only 50%. The additional 30% reduction – which is still significant – is achieved by increasing average connectivity from 50% to 100%. From a practical perspective, these trends allow trades to be made between gains in covariance and the costs of implementing highly-robust networks.

The investigation of trends in covariance, however, must be accompanied by the impacts of network connectivity and the Local Fusion Graph method on state estimation. Figure 26 through Figure 28 present the results of average state estimation versus time for the three network architectures. As with the average covariance results, the state estimates of the Local Fusion Graph model at 0% and 100% connectivity are the same as those of the closed form algorithms and confirm the soundness of the Local Fusion Graph approach.

The state estimation results show that average estimation capabilities do not necessarily improve with increasing network connectivity. The estimates from the ad hoc and broadcast architectures show that estimates under 0% network connectivity are more accurate than at 25% connectivity for some time steps. Similarly, results from the cyclical architecture show that estimates calculated at 0%, 25%, and 50% connectivity are relatively consistent.

Further insight into these estimation trends can be measured by the RMS error of the estimates. The RMS error values are calculated according to Equation (39) and are

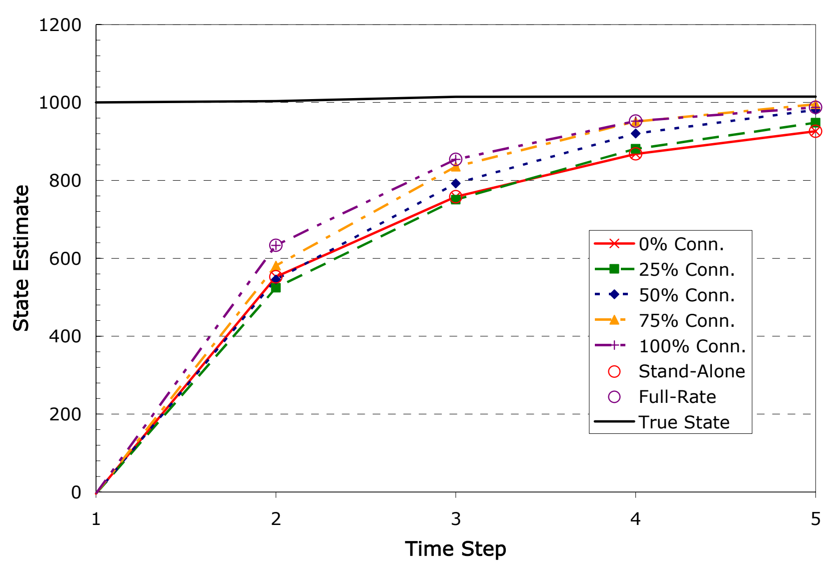

Figure 26: State Estimate vs. Time for Various Network Connectivity Values in an Ad Hoc Network under Transient Estimation Conditions.

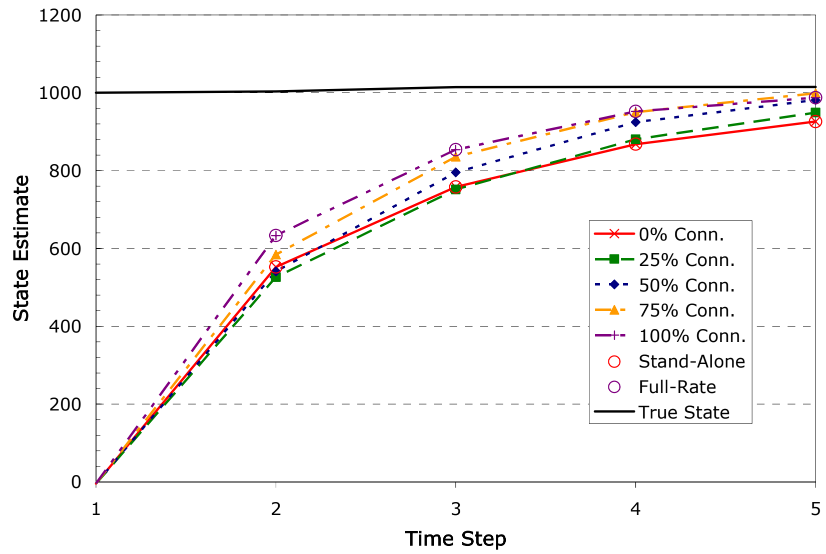

Figure 27: State Estimate vs. Time for Various Network Connectivity Values in a Broadcast Network under Transient Estimation Conditions.

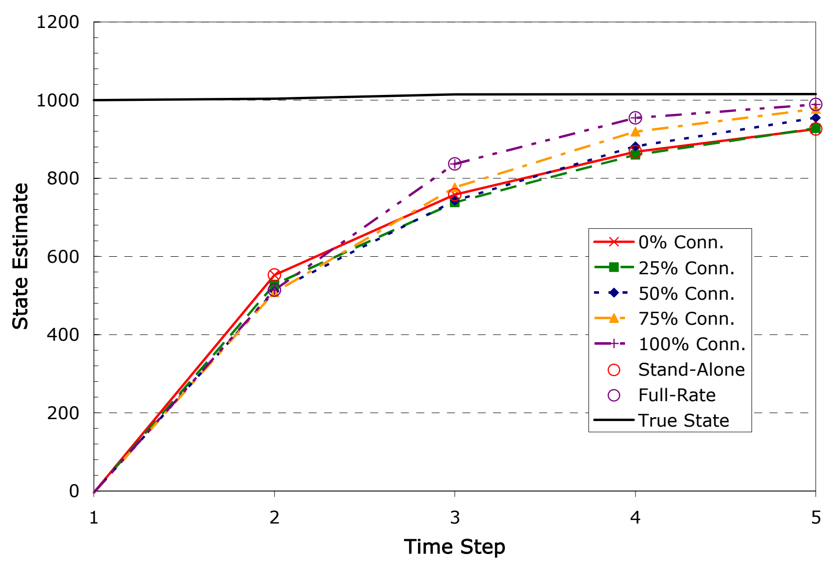

Figure 28: State Estimate vs. Time for Various Network Connectivity Values in a Cyclical Network under Transient Estimation Conditions.

collated for each connectivity level and each time step. The results for the RMS error versus time for each connectivity level are shown in Figure 29 through Figure 31. Overall the RMS errors show a general consistency with absolute state estimates shown in Figure 26 through Figure 28 with some noted exceptions. Like the state estimate, a general increase in state estimation RMS error results from increased network connectivity. Likewise the RMS error data indicates that the 25% connectivity case results in greater estimation errors than that of the 0% connectivity case. Additionally, the RMS errors for the 0%, 25%, and 50% connectivity cases under the cyclic architecture are shown to be quite similar in magnitude for each of the five time steps.

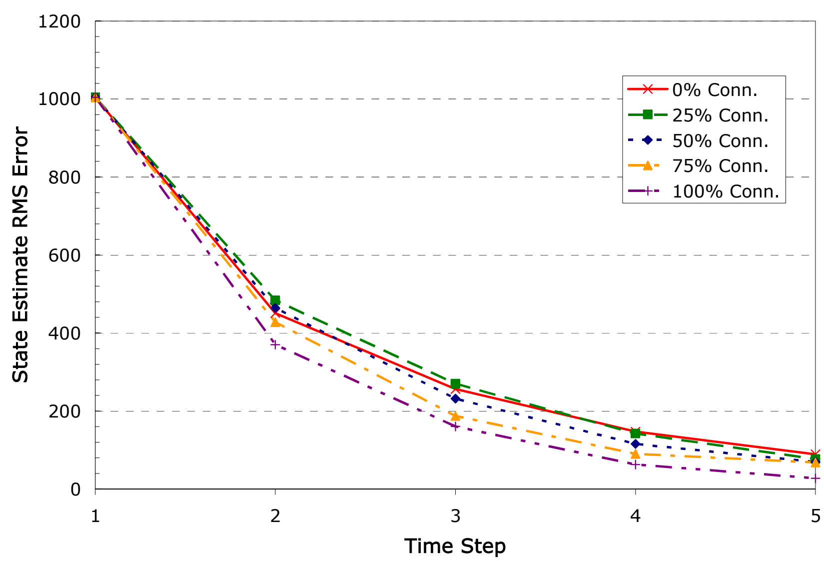

Figure 29: State Estimate RMS Average Error vs. Time for Various Network Connectivity Values in an Ad Hoc Network under Transient Estimation Conditions.

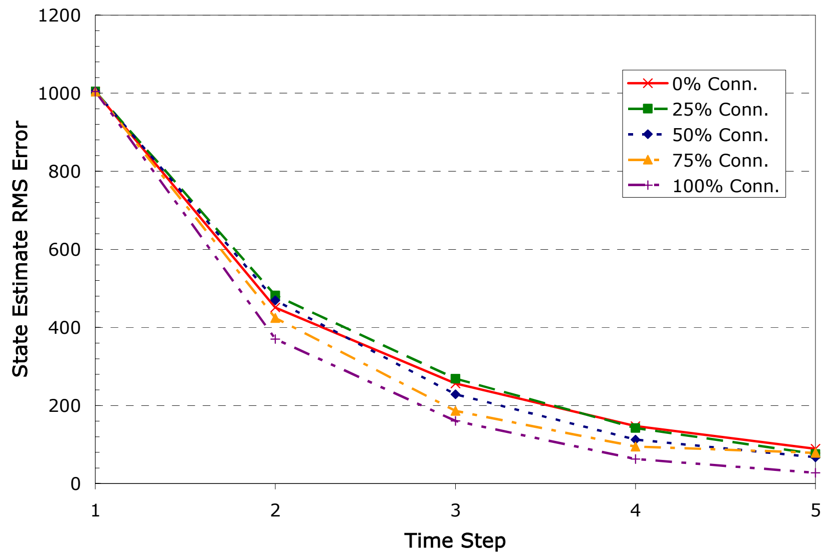

Figure 30: State Estimate RMS Average Error vs. Time for Various Network Connectivity Values in a Broadcast Network under Transient Estimation Conditions.

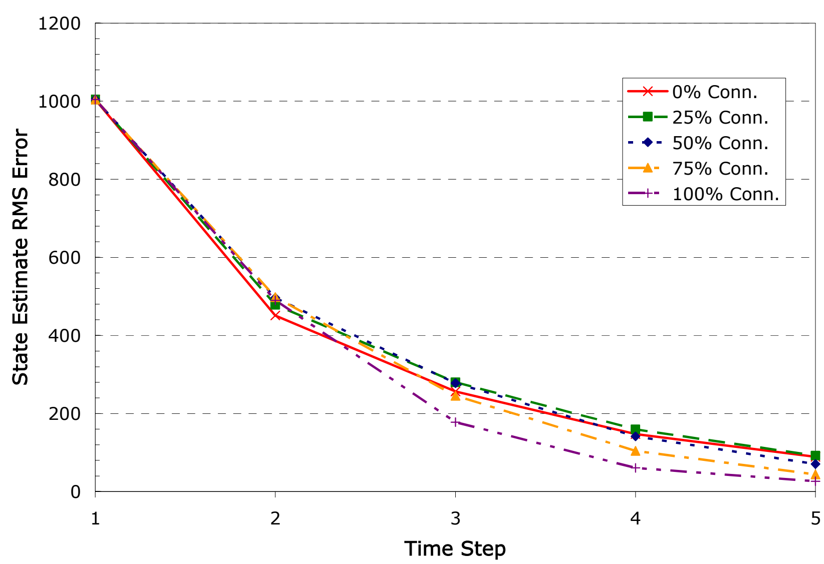

Figure 31: State Estimate RMS Average Error vs. Time for Various Network Connectivity Values in a Cyclical Network under Transient Estimation Conditions.

To better understand the relationship between average network connectivity and estimation performance, the RMS errors are averaged across the last four time steps for each connectivity level as shown in Figure 32. The initial time step was excluded from the data to reduce bias resulting from the initial state estimates chosen for the scenario. The data shown by the graph present a rather interesting result. Specifically, the time-averaged RMS errors are shown to initially increase at low non-zero average connectivity levels relative to 0% connectivity. The time-averaged RMS errors are greatest at 25% connectivity for the ad hoc and broadcast cases, which are also noted to be nearly identical. For the cyclic communications architecture, the averaged RMS errors at 50% connectivity continue to be greater than that at 0% network connectivity. Beyond the

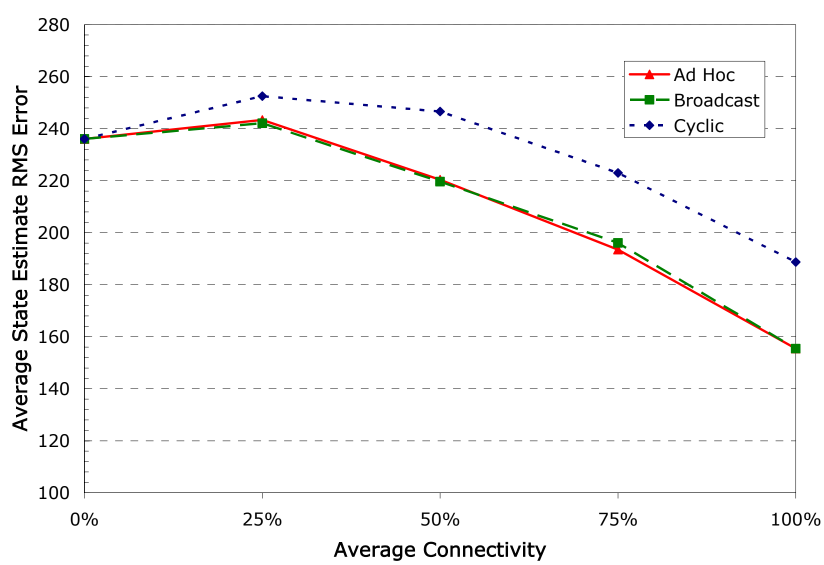

Figure 32: Time-Averaged RMS Estimation Error vs. Network Connectivity in Ad Hoc, Broadcast, and Cyclical Networks under Transient Estimation Conditions.

peak RMS error at 25% connectivity, the estimation performance improves at an increasing rate with incremental growth in average network connectivity.

Additional simulations were performed to better characterize the trend in average RMS error at low network connectivity levels. The data shown in Figure 33 demonstrate the characteristics of time-averaged RMS error under the same conditions as Figure 32, but at five increments between 0% and 50% connectivity. The peak RMS error actually occurs at or near 12.5% connectivity for ad hoc and broadcast architectures but occur at 25% for the cyclical architecture.

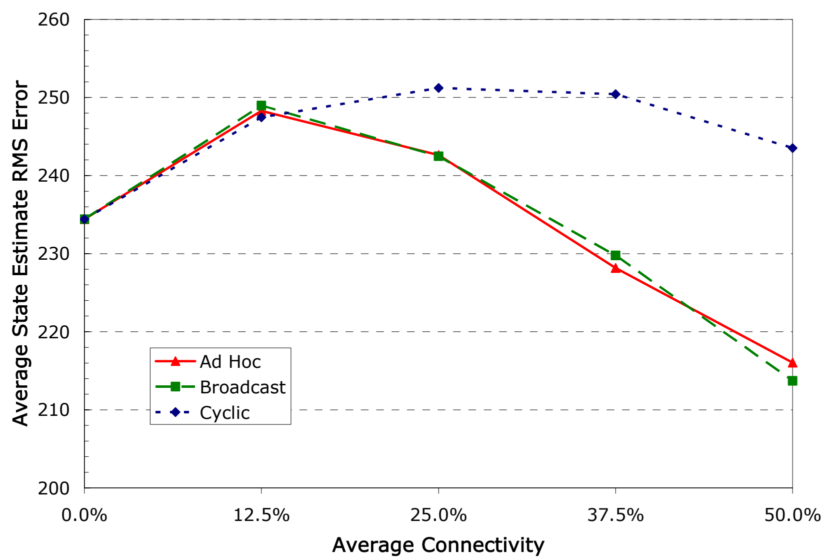

Figure 33: Time-Averaged RMS Estimation Error vs. Network Connectivity Between 0% and 50% under Transient Estimation Conditions.

Further exploration of the KF algorithm, fusion equations, and simulations using a range of scenario parameters provides insight into the general behavior of the average RMS trends. The key to understanding the phenomena is found from individual simulation runs rather than the data averaged across all 1000 Monte Carlo runs. For individual simulation runs, the covariance for a fusion agent in the randomly connected networks fluctuates based on how often it communicates with other agents as well as the prior history of the data it inherits. Communication (e.g. fusion) events lower the covariance while time steps without fusion events will often increase it (or not decrease it as rapidly). Using Figure 19 as an example, the 75% connectivity case has fusion events at the first three time steps, but not a the fourth. Thus the covariance increases at k=4. Similarly, the 25% connectivity case has a fusion event at k=1 but not at k=2, thus the covariance reduction is not as significant as that at k=1.

In these cases, the covariance does not match the changing network conditions. If a fusion event at one time step is followed by a situation where no fusion events occur at the next time step, the covariance may then be "optimistic" for the stand-alone estimation. The reduced covariance produces a smaller filter gain and the measurement (innovation) therefore has a lesser impact on the state estimate than would be expected. Thus agents with no connectivity may have a more accurate state estimate than agents with sporadic connectivity if the measured state is more accurate than the predicted state.

The impact of the connectivity characteristics vary based on the ratio between process and measurement noises. The ratio of process to measurement noise used for the simulations is 1:4. As an excursion, three different noise ratios were used for the transient estimation scenario and the resulting average RMS errors are presented in Figure 34 and Figure 35. As shown, increasing process noise relative to measurement noise produces the increased inaccuracies at low network connectivity levels.

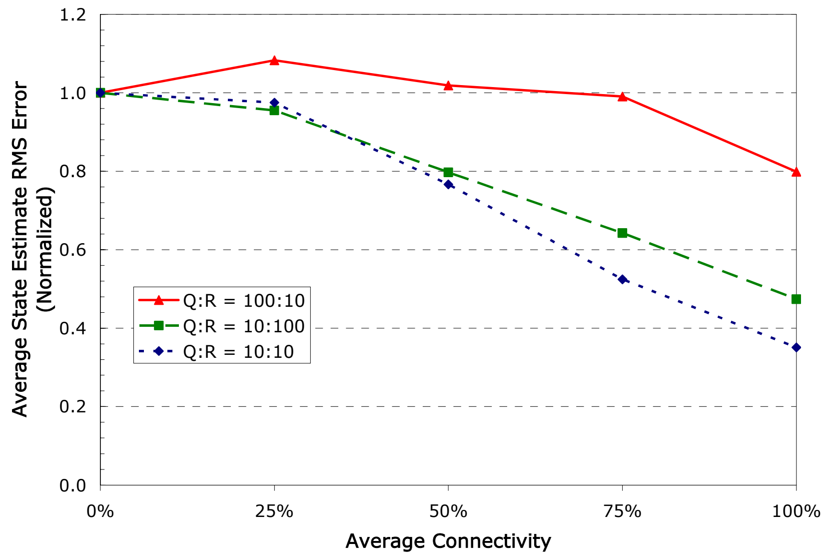

Figure 34: Time-Averaged RMS Estimation Error vs. Network Connectivity for Various Noise Ratios in an Ad Hoc or Broadcast Communications Network.

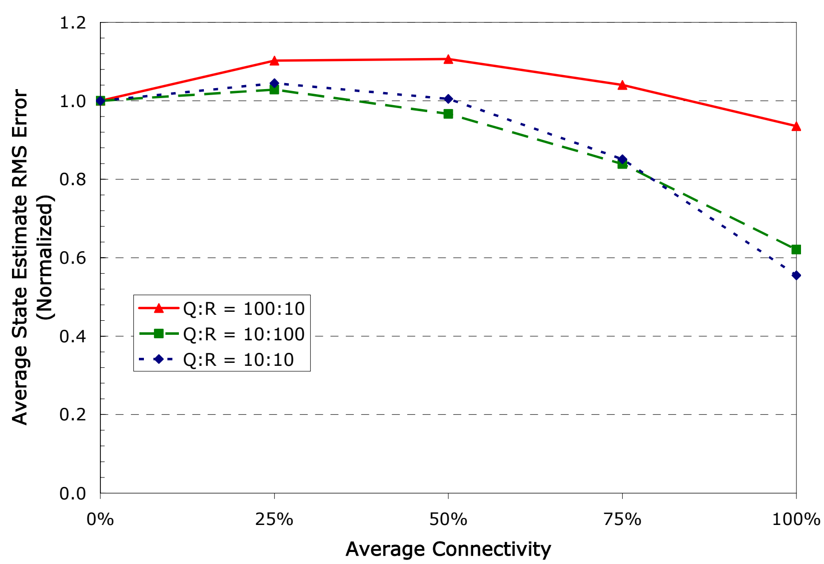

Figure 35: Time-Averaged RMS Estimation Error vs. Network Connectivity for Various Noise Ratios in Cyclical Communications Network.

Overall, the data produced by the computer model provide significant insight regarding estimation capabilities relative to network connectivity under transient estimation conditions. First and foremost, the Local Fusion Graph methodology is seen to be consistent with known estimation solutions for deterministic network connectivity. Additionally, reductions in covariance are found to be non-linear with respect to network connectivity. State estimates, however, demonstrate that estimation errors for networks with low non-zero average connectivity are larger than that of stand-alone estimation. Estimation performance then increases as average network connectivity exceeds 12.5%. Also of significance is the finding that the average estimation capabilities under the ad hoc and broadcast network architectures are essentially identical. While the limited data presented here is insufficient to declare that many of the observed estimation characteristics and trends are general to all estimation scenarios, the data produced for steady-state estimation conditions in the following section provide further insight.

3.2.2 Analysis of Local Fusion Graphs under Steady-State Estimation Conditions

As previously described, the simulation of steady-state estimation conditions provides data on the characteristics of distributed fusion in networks with various degrees of connectivity while the estimates are within the bounds of achievable accuracy. The analysis also provides further insight into the previous section's findings, thus helping to discern general trends from scenario-dependent results. As such, the analysis follows the same line of inquiry as the analyses of transient conditions, which investigated covariance, state estimates, and state estimate RMS errors averaged across 1000 Monte Carlo runs.

Figure 36 through Figure 38 depict the average covariance under steady-state estimation conditions. Because the initial covariance is close to achievable levels, the variances are little changed with respect to time. As with transient conditions, however, significant changes in covariance with respect to network connectivity are found. Figure 39 through Figure 41 demonstrate the relationship of covariance versus network connectivity for each of the five time steps. Excluding the first time step, which is influenced by initial conditions, the trends in covariance reduction with increasing network connectivity are very similar for each to the remaining four time steps. As postulated in the analysis for transient conditions, it is apparent that the relationship between covariance and connectivity converge to a steady-state over time. That fact is further supported by the equivalence shown in covariance reductions achieved at k=5 under transient conditions in Figure 23 through Figure 25 with those achieved under steady-state conditions in Figure 39 through Figure 41.

Unlike the state estimates produced under transient estimation conditions, the state estimates found under steady-state conditions and presented in Figure 42 through Figure 44 are characterized by rapid but small variations in response to similar changes in the true state. Given that the same state and measurement models were used across all Monte Carlos and all network connectivity cases, the figures demonstrate that greater degrees of network connectivity result in more rapid responses to differences between the true and estimated states as well as to changes in the true state. Increased response to differences between true state and the estimate is shown in the state estimate graphs as

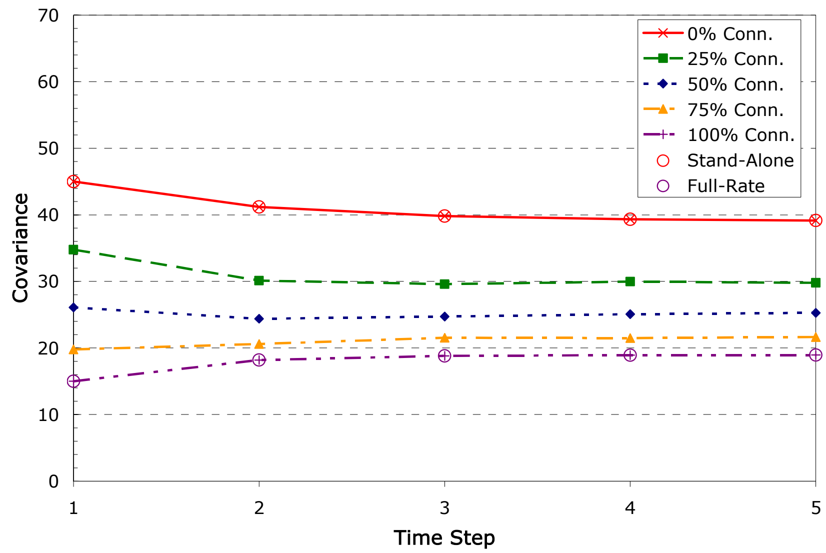

Figure 36: Covariance vs. Time for Various Network Connectivity Values in an Ad Hoc Network under Steady-State Estimation Conditions.

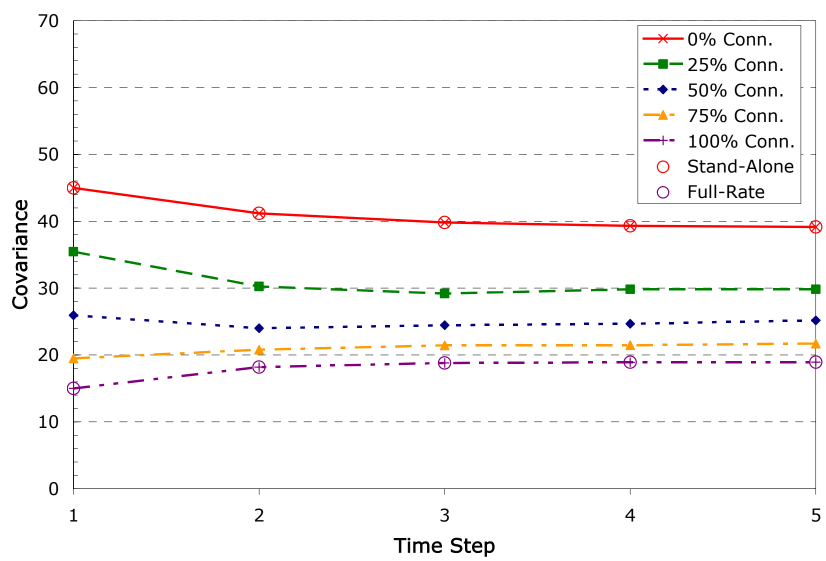

Figure 37: Covariance vs. Time for Various Network Connectivity Values in a Broadcast Network under Steady-State Estimation Conditions.

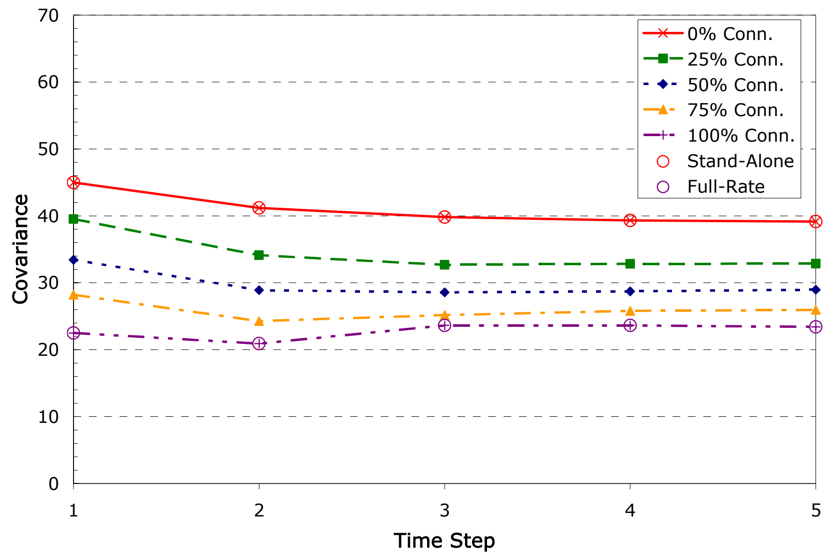

Figure 38: Covariance vs. Time for Various Network Connectivity Values in a Cyclical Network under Steady-State Estimation Conditions.

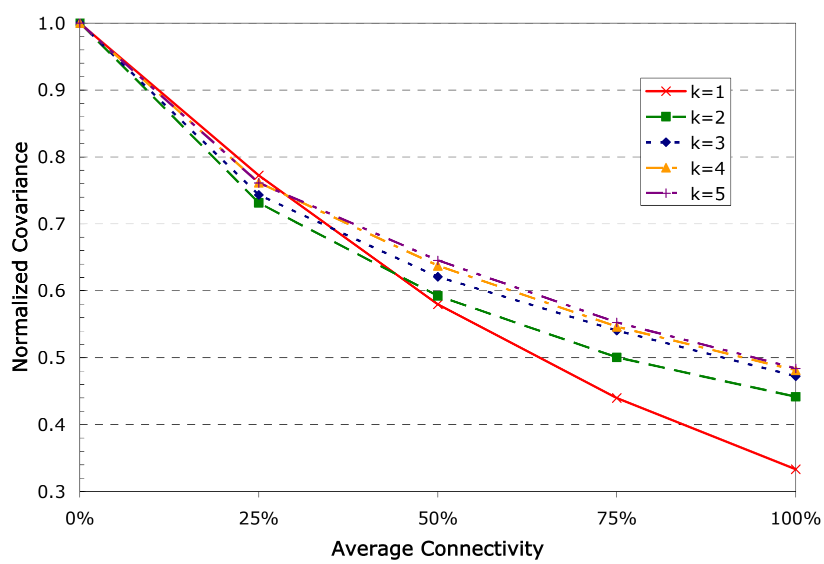

Figure 39: Normalized Covariance vs. Average Network Connectivity for Each Time Step in an Ad Hoc Network under Steady-State Estimation Conditions.

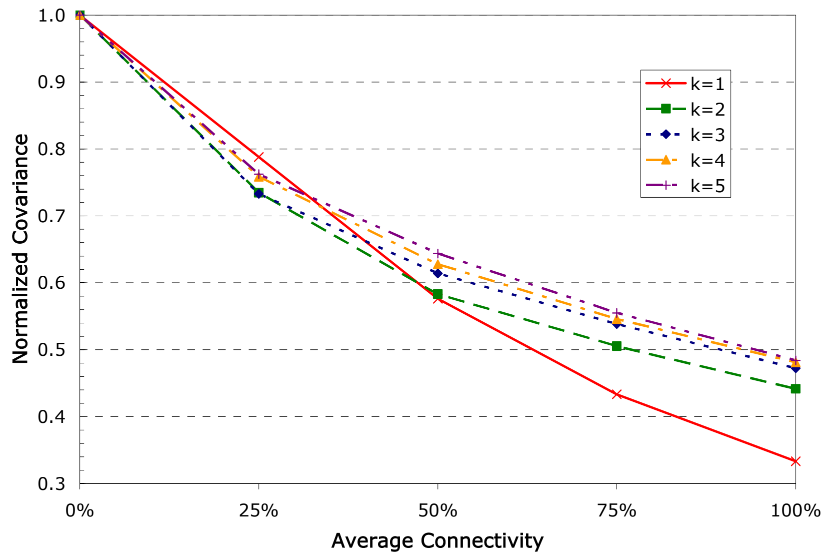

Figure 40: Normalized Covariance vs. Average Network Connectivity for Each Time Step in a Broadcast Network under Steady-State Estimation Conditions.

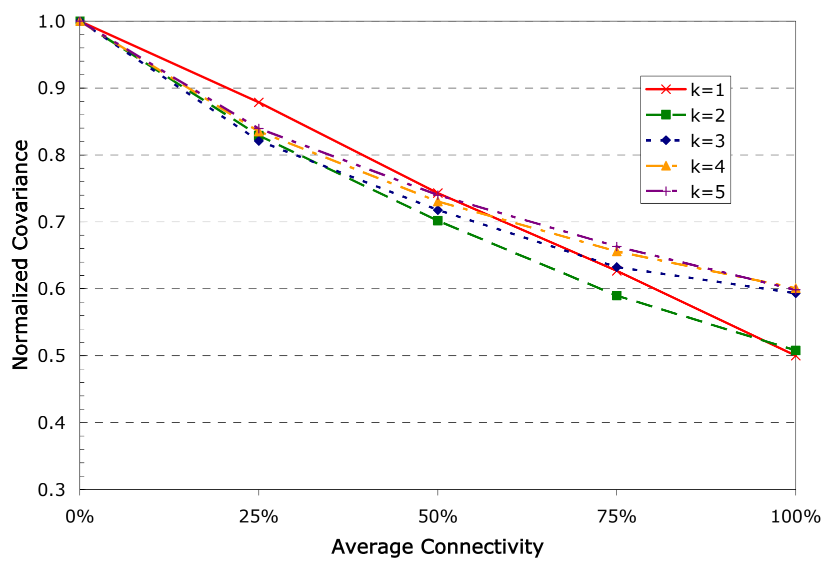

Figure 41: Normalized Covariance vs. Average Network Connectivity for Each Time Step in a Cyclical Network under Steady-State Estimation Conditions.

well as the RMS average error data shown in Figure 45 through Figure 47. The RMS error is shown to decrease more rapidly with greater network connectivity where the true state maintains a position in the direction of the state estimation trajectory. That condition is essentially the same trait that is found under transient estimation conditions.

The tendency for increased response to changes in true state with increased network connectivity are shown by the estimate changes from k=1 to k=2. In this situation, all connectivity values have the same initial estimate but respond to the true state differently despite the fact that the same measurement is used for all connectivity cases. For the scenario data presented here, the RMS error decreases for 75% and 100% connectivity, but increases for the other three cases. Furthermore, the RMS error for 25%

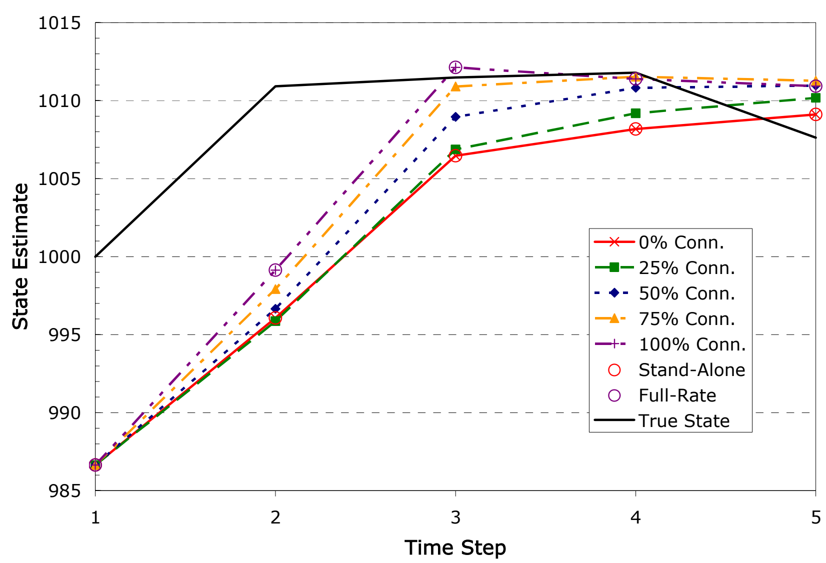

Figure 42: State Estimate vs. Time for Various Network Connectivity Values in an Ad Hoc Network under Steady-State Estimation Conditions.

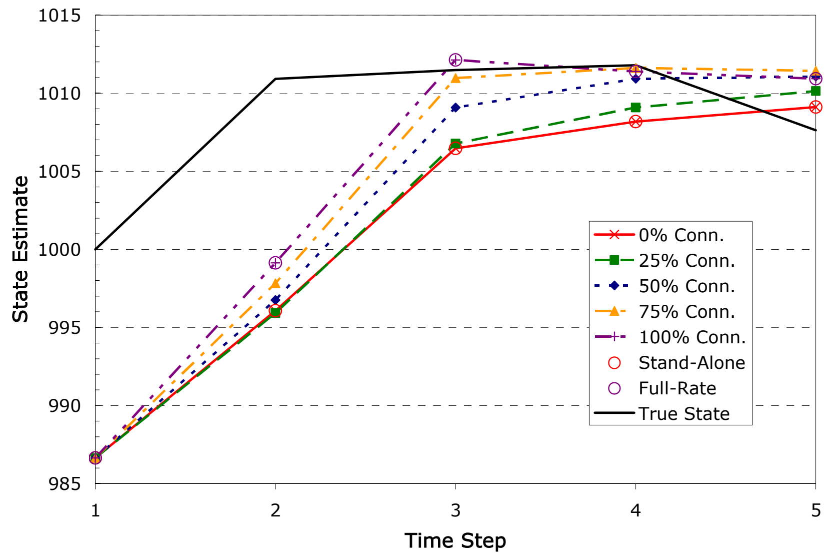

Figure 43: State Estimate vs. Time for Various Network Connectivity Values in a Broadcast Network under Steady-State Estimation Conditions.

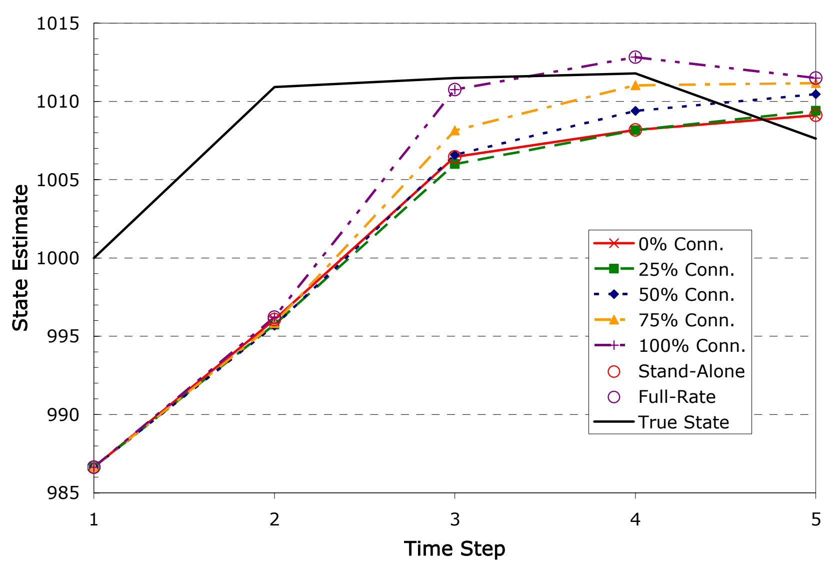

Figure 44: State Estimate vs. Time for Various Network Connectivity Values in a Cyclical Network under Steady-State Estimation Conditions.

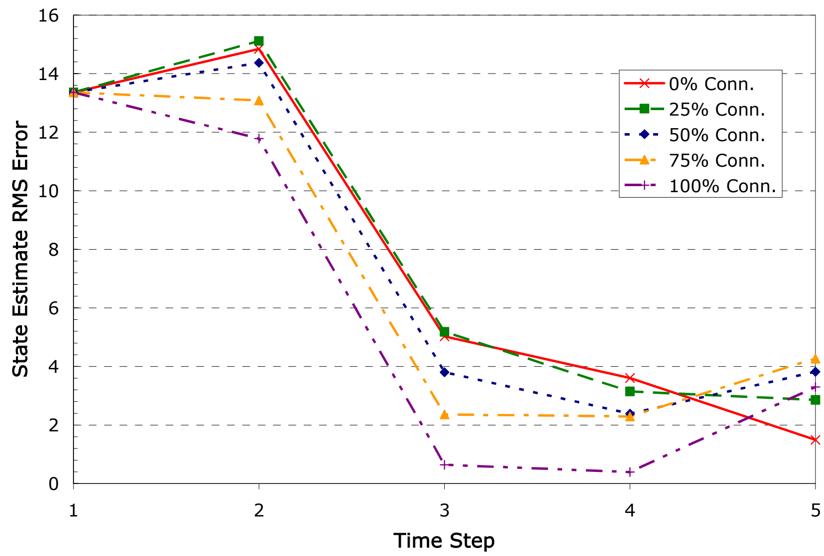

Figure 45: State Estimate RMS Average Error vs. Time for Various Network Connectivity Values in an Ad Hoc Network under Steady-State Estimation Conditions.

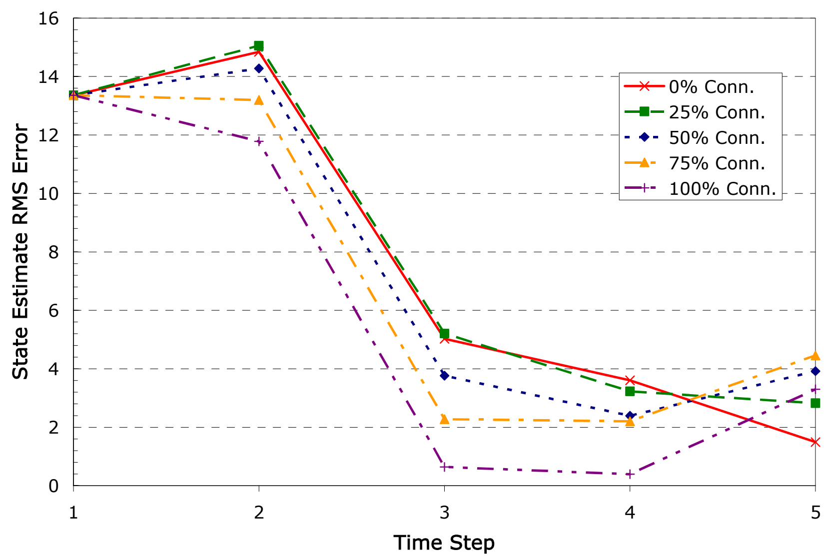

Figure 46: State Estimate RMS Average Error vs. Time for Various Network Connectivity Values in a Broadcast Network under Steady-State Estimation Conditions.

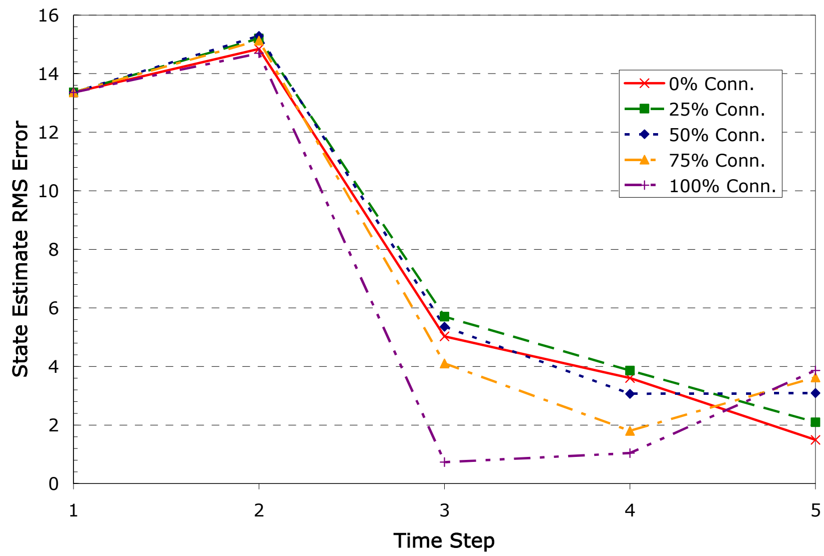

Figure 47: State Estimate RMS Average Error vs. Time for Various Network Connectivity Values in a Cyclical Network under Steady-State Estimation Conditions.

connectivity is greater than that of 0% connectivity, which is consistent with the findings discussed for transient estimation conditions.

It is cautioned, however, that better connectivity, does not guarantee better estimation results at all time steps for the other connectivity levels. Changes in state such as that shown for k=5 combined with measurement noises may create temporary fluctuations of state estimation accuracy with respect to network connectivity. That effect is common for conditions modeled by the steady state scenario. The statistic of time-averaged RMS is therefore an important metric for evaluating estimation capabilities with respect to average network connectivity.

The characteristics of state RMS average error versus network connectivity are very similar between the steady-state and transient estimation conditions. While the scales of the errors are very different, the change in RMS average error with network connectivity under steady-state conditions shown in Figure 48 mirrors the trends found for transient conditions demonstrated in Figure 32. The initial increase in average RMS error is found as network connectivity increases from 0% to 25%, followed by an increasingly rapid drop-off for the remainder of the connectivity values.

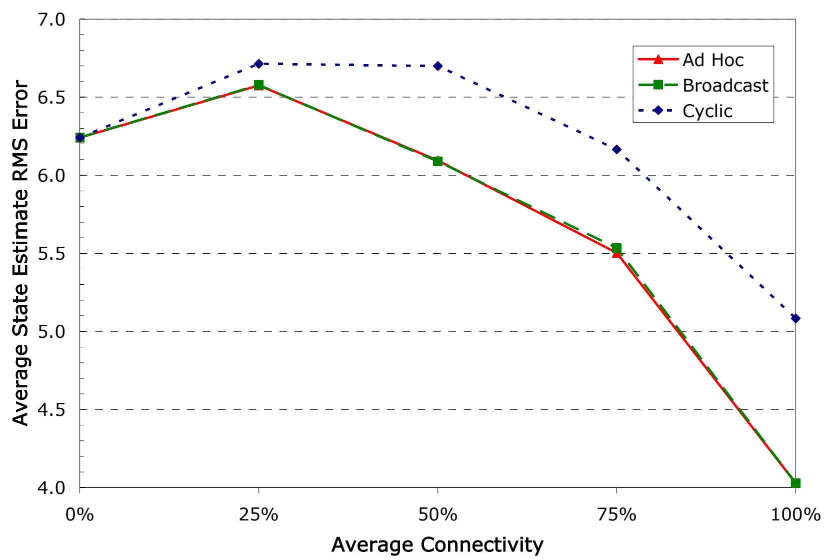

Figure 48: Time-Averaged RMS Estimate Error vs. Average Network Connectivity in Ad Hoc, Broadcast, and Cyclical Networks under Steady-State Estimation Conditions.

While improved capability is the ultimate goal of a distributed data fusion system, the communications requirements for supporting the exchange of data must be considered. As discussed in Chapter 1, systems that are reliant upon wireless communications are encumbered with many issues that limit data transfer capabilities. The communications needs for the Local Fusion Graph method are of particular importance since they are intended to be used for distributed wireless systems. The next section provides an assessment of data exchange requirements for the Local Fusion Graph method to provide some basic insight into communications needs.

3.3 Communications Considerations for the Local Fusion Graph Method

Communications in support of distributed information fusion nodes utilize resources in terms of energy, network capacity, and spectrum when exchanging data. These resources are very limited in many cases and are therefore very valuable for wireless networks. Thus, understanding the impact of the Local Fusion Graph method on communications requirements is vital.

The derivation example in Chapter 2 shows that the size of the local fusion graphs grow according to $O(n \cdot k)$, making the transmission of the entire graph impractical. Optimally, it is desirable for agents to only exchange information that is new to the receiving agent(s) rather than exchanging significant amounts of data that will be discarded because it is redundant. While this optimal condition is unlikely to be practical, a number of techniques and communications protocols can be employed to reduce communications loads.

The most obvious technique is to use a sliding time window to determine which data is included in each transmission. In this technique, agents determine how much of the time history is required to maintain a desired accuracy in the estimate. The data messages therefore contain only the information required to achieve the accuracy threshold. A second simple technique is for agents not to send data to an agent that originated the data. For example, if agent *A* maintains estimates from agent *B* in its local fusion graph, it need not send data related to agent B back to agent B since each agent inherently preserves its own information locally.

In addition to these techniques, communications protocols can be designed to minimize communications requirements. For the first protocol, agents exchange data control messages that allow the sender of the fusion data to determine which data is required at the receiving agent. The messages from the intended recipient of the fusion data would contain the unique identifiers of the most recent nodes in its local fusion graph. The provider of the fusion data would then only send data associated with more recent estimates. Note that this technique works similar to packet acknowledgement protocols and is only applicable to point-to-point communications links.

A protocol that is applicable to wireless broadcast networks is for all agents to glean information from a common data exchange channel. In this protocol, an agent broadcasts its fusion messages on a channel that is shared among a group of agents. All other agents within reception range demodulate the message and check it for new data. By monitoring the channel, agents can gain an understanding of what data has been sent to the group in past transmissions and assemble future messages to exclude the previously-transmitted data from its message. From a practical perspective, each agent may be in range of multiple subnets, multiple sections of a mesh network, or may be a mobile agent and thus hear multiple messages, each with different data. Transmitting a set of data that is new to all nodes within reception range is therefore not likely to be possible. The nodes, however, may be able to construct messages that contain the most new data and smallest duplication of data.

The best method for a given fusion network is dependent upon the capabilities of the agents' communications systems, the capacity of the network, and the performance requirements of the fusion process. Fusion processes that operate in extremely fast cycles may need to utilize fixed network topologies with known message contents or must limit their accuracy by employing heuristics. Other fusion processes may be able to utilize communications protocols to manage the amount of data transmitted. In practice, the protocol implemented in the network will be designed to fit the communications capabilities. In fact, there is likely to be no single best protocol that fits all distributed fusion networks. Instead, different techniques will be effective under different assumptions and network designs. The methods presented here are not intended to be an exhaustive list, but ones that illuminate some of the fundamental considerations.

While designing communications protocols for the efficient exchange of fusion data is beyond the scope of this effort, a basic assessment of data exchange needs is in order. The next section examines the data exchange needs of the Local Fusion Graph method to determine the minimum amount of data that must be exchanged between agents for completely efficient communications. Section 3.3.2 then compares communications needs with respect to estimation performance for the Local Fusion Graph and naïve fusion methods.

3.3.1 Examination of Local Fusion Graph Communications Requirements

Given that the analysis of multiple communications protocols and a determination of a "best" technique are beyond the scope of this thesis, communications requirements were assessed in terms of the amount of new data acquired by the receiving agent. This approach provides an assessment of relative communications data rate requirements with respect to network connectivity. Because the new data acquired by an agent depends on the current and prior data sets, it was found that the initial three to five time steps bias the results due to the shorter time history of the early local fusion graphs. Therefore, the simulation model was run for 10 time steps at 1000 Monte Carlos, and the first five time steps were excluded from the data sets.

The communications requirements are defined here in terms of the number of time-agent nodes in the fusion graph that must be sent in a message at each time step by an agent in the network. The metric is given as nodes per message, abbreviated as nodes/msg. It is important to note that the data transmitted in the message includes the local estimation data associated with the respective node in the local fusion graph as well as the relevant pointers connecting the node to the rest of the graph.

The histograms of the average message size required for each agent in terms of nodes/msg are presented in Figure 49 through Figure 51 for various degrees of network connectivity in ad hoc, broadcast, and cyclical communications architectures. Only the distributions for non-deterministic network connectivity levels are presented; the distributions for 0% and 100% connectivity are merely unit impulses at 0 and 2 nodes/msg, respectively.

The distributions for each successive increase in network connectivity are shown to change with network connectivity. As demonstrated in Figure 52 and Table 3, the mean message size shifts from 0 nodes/msg to 2 nodes/msg with increasing network

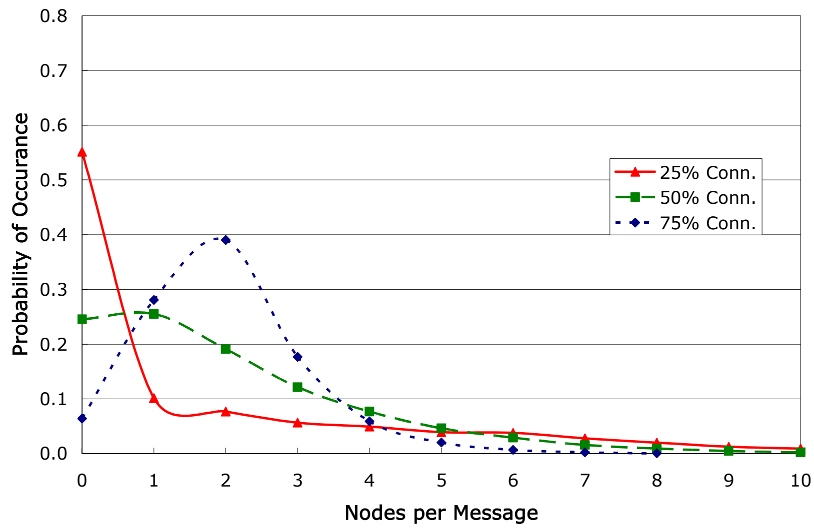

Figure 49: Histogram of Message Size for the Local Fusion Graph Method in an Ad Hoc Communications Network.

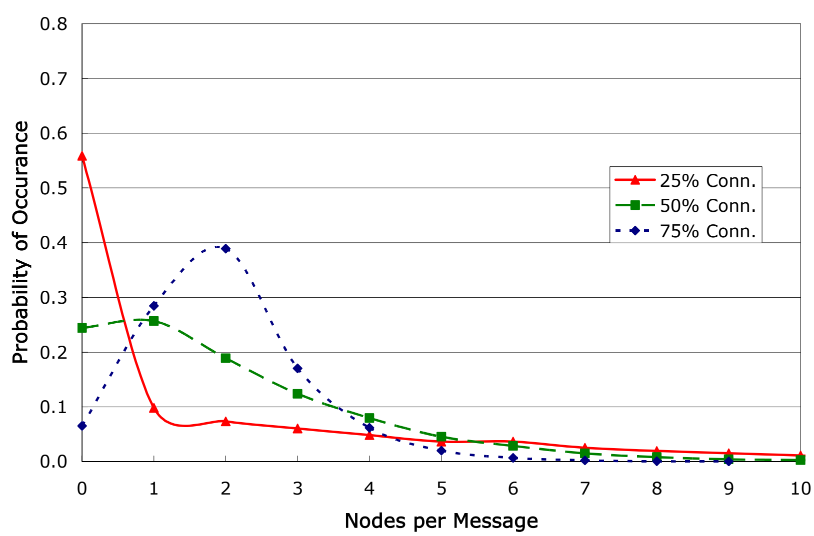

Figure 50: Histogram of Message Size for the Local Fusion Graph Method in a Broadcast Communications Network.

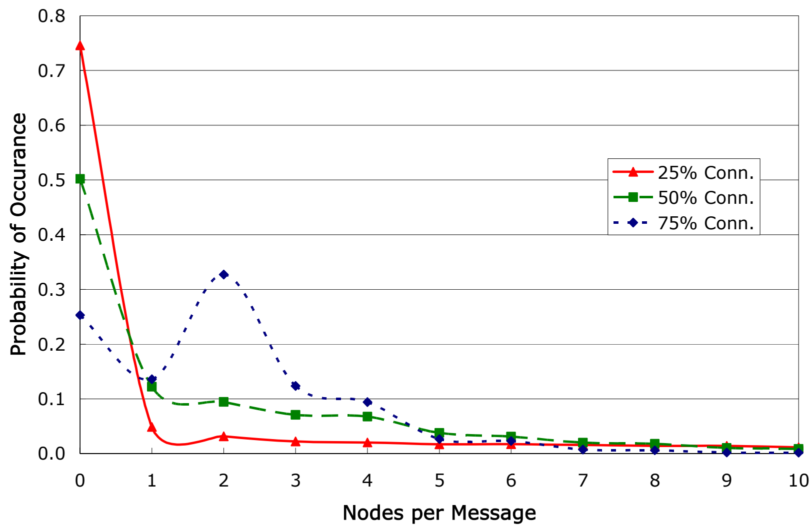

Figure 51: Histogram of Message Size for the Local Fusion Graph Method in a Cyclical Communications Network.

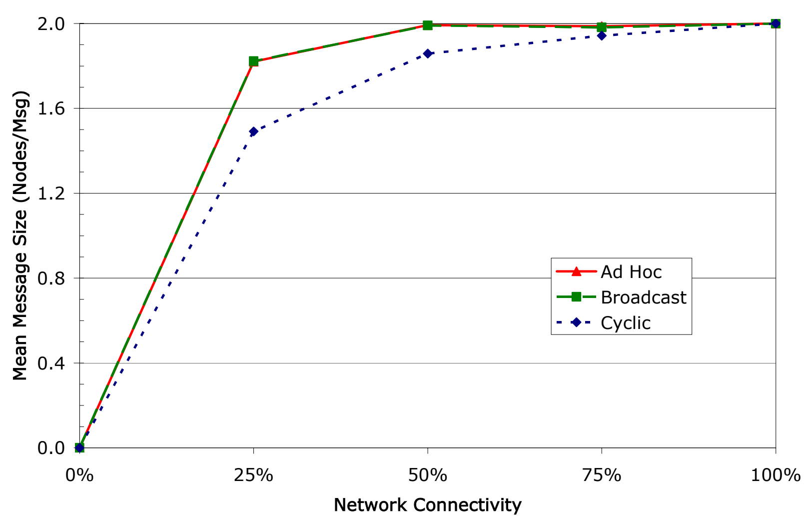

Figure 52: Network Connectivity vs. Average Message Size for the Local Fusion Graph Method.

Table 3: Message Size (Nodes/Msg) Statistics for Local Fusion Graph Communications.

Conn.	Ad Hoc			Broadcast			Cyclic		
	Mean	Max	Std Dev	Mean	Max	Std Dev	Mean	Max	Std Dev
0%	0.000	0	0.000	0.000	0	0.000	0.000	0	0.000
25%	1.820	22	2.843	1.823	24	2.896	1.491	23	3.432
50%	1.992	16	1.974	1.991	18	1.970	1.858	21	2.766
75%	1.988	8	1.146	1.982	9	1.154	1.943	11	1.694
100%	2.000	2	0.000	2.000	2	0.000	2.000	2	0.000

connectivity. Two primary observations are made from this data. First, the mean message size is always less than that of a fully-connected network. Second, the increase in mean message size is rapid, with an average of 1.8 nodes/msg at 25% network connectivity. Thus there is little difference in the average message size in the three network connectivity values modeled here.

While the average message size is comparable for the three cases examined, the maximum message size changes significantly with an inverse relationship relative to network connectivity. The result is fairly intuitive: as agents exchange data less often, the average message size decreases due to reduced number of messages being sent; When agents do communicate, however, message sizes are greatly increased because they must contain greater amounts of data to complete the fusion graph at each agent. Thus there is an inverse relationship between the peak and average message sizes required to support the Local Fusion Graph method.

Cumulative probability distributions along with mean and maximum message size data provide further insight into message sizes. The distributions are shown in Figure 53 through Figure 55 and can have a couple interpretations. For variable message size communications, the figures show the probability that messages are of the given size or smaller. As an example, Figure 53 shows that approximately 90% of the messages in an ad hoc network with 75% connectivity would be no more than 3 nodes/msg. For fixed message size communications, the figures indicate the probability that messages of the given size are sufficient for exchanging the unique data between agents. Thus Figure 53 shows that a message size of 3 nodes/msg is sufficient for 90% of the data exchanges in an ad hoc network with 75% connectivity.

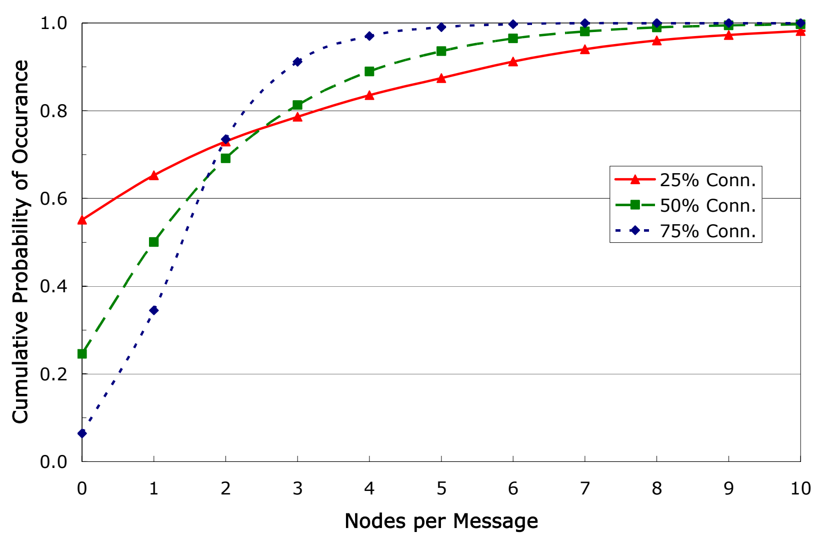

Figure 53: Cumulative Distribution for Local Fusion Graph Message Size in an Ad Hoc Communications Network.

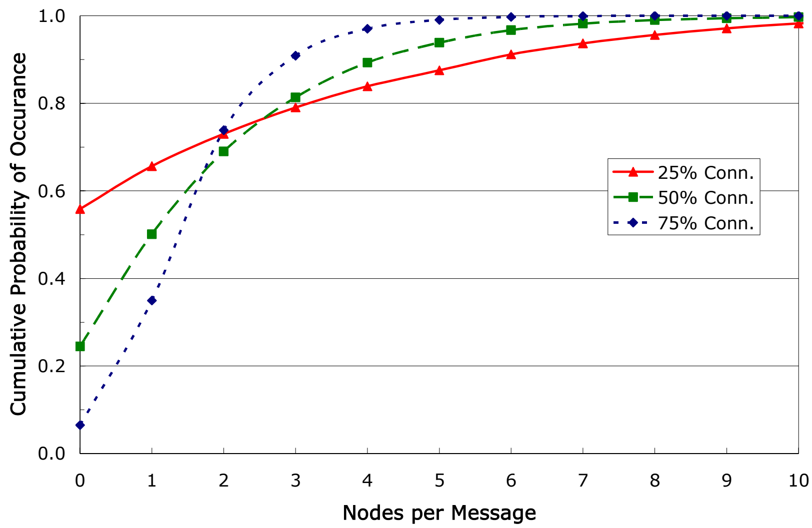

Figure 54: Cumulative Distribution for Local Fusion Graph Message Size in a Broadcast Communications Network.

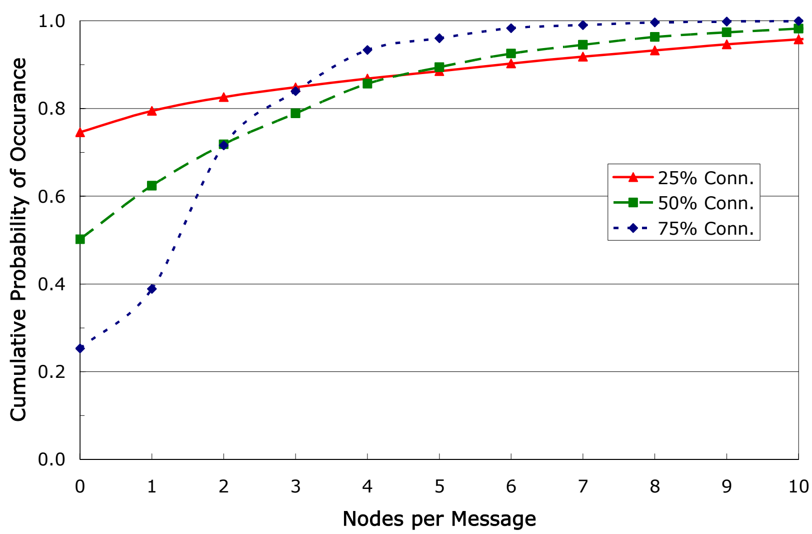

Figure 55: Cumulative Distribution for Local Fusion Graph Message Size in a Cyclical Communications Network.

Two caveats are attached to the communications assessments. First, the data represents a lower bound on the statistics for actual implementations. The data presented in the analysis is derived from the number of new nodes added to an agent's fusion graph at each time step. If the sending agent doesn't know exactly which data is needed by the receiving agent, then it will likely send duplicative data to increase the probability of providing sufficient data and thus increase the data contained in each message. Furthermore, the sending agent would need to know *which* node data needs to be sent. The analysis here simply quantified the number of nodes, but made no distinction as to their location in the local fusion graph. Further analysis is required to determine how agents can provide all needed data with the least amount of unnecessary data.

The second caveat is that the data does not indicate how the probability distributions scale with the number of nodes in the network. Attempts were made to characterize the communications requirements distributions produced by the Local Fusion Graph algorithm, but the distributions of the simulation data could not be accurately fit to a common distribution. For the first simulation time step, the probability distribution is given by the binomial:

$$\beta(m; n-1, \mu) = \binom{n-1}{m} \mu^m (1-\mu)^{(n-1)-m} \tag{40}$$

where μ is the average probability of message delivery (e.g., the average network connectivity), m is the expected number of nodes in the message, and n is the total number of fusion agents in the network. At subsequent steps, however, the formulation must also incorporate the probability of each possible combination of messages at all prior time steps. Thus the distribution is the binomial for the current time step plus additional binomial terms that characterize the probability of all possible prior events. This line of reasoning is supported by the consistency between the simulation results of 0 nodes/msg at k=1 shown in Figure 49 through Figure 51 and $\beta(0;n\text{-}1,\mu)$, given as:

$$\beta(0; n-1, \mu) = (1-\mu)^{n-1} \tag{41}$$

While a complete statistical characterization is yet to be performed, the development of a stochastic fusion method in Chapter 4 sheds further light on the communications characteristics.

3.3.2 Comparison of Local Fusion Graph and Naïve Fusion Communications

The assessments in the prior section provide valuable insight for communications in the Local Fusion Graph method. Further observations can be provided by a comparison with other distributed fusion methods. Because naïve fusion is the most common method found in literature and is adaptable to ad hoc communications networks, it is chosen for comparison with respect to data exchange needs in light of the accuracy of the resulting estimates.

In a naïve fusion algorithm, fusion agents exchange only the most recent estimates with other fusion agents without regard for possible data dependencies [32]. Thus the fusion equations become:

$$\hat{\mathbf{x}}_j(k) = \mathbf{P}_j(k)\left(\sum_{i=1}^{n}\mathbf{P}_i^{-1}(k \mid k)\hat{x}_i(k \mid k)\right) \tag{42}$$

$$\mathbf{P}_j(k) = \left(\sum_{i=1}^{n}\mathbf{P}_i^{-1}(k \mid k)\right)^{-1} \tag{43}$$

Figure 56 and Figure 57 present the results for transient estimation conditions, and Figure 58 and Figure 59 present the results for the steady-state estimation conditions. The trends in covariance are similar to those of the Local Fusion Graph method, with non-linear reductions relative to increasing time and network connectivity levels. Naïve covariance, however, decreases more rapidly. It is shown in Figure 56 that the reductions in naïve covariance exceed known solutions. Similarly, the naïve state estimates are

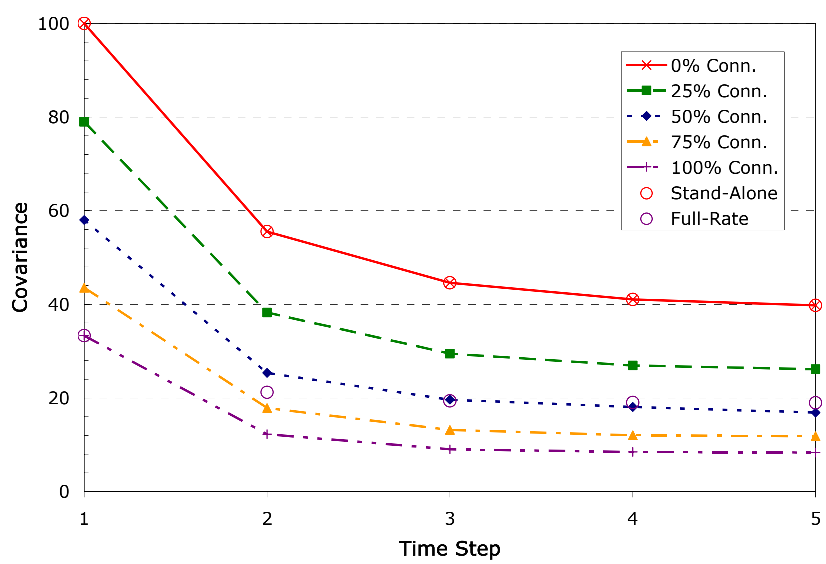

Figure 56: Naïve Communications Covariance Estimate versus Time for an Ad Hoc Communications Architecture under Transient Estimation Conditions.

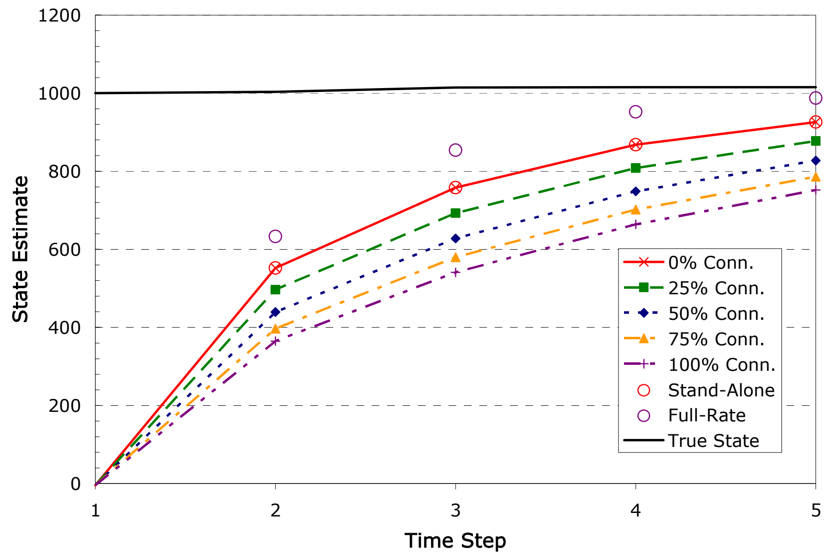

Figure 57: Naïve Communications State Estimate versus Time for an Ad Hoc Communications Architecture under Transient Estimation Conditions.

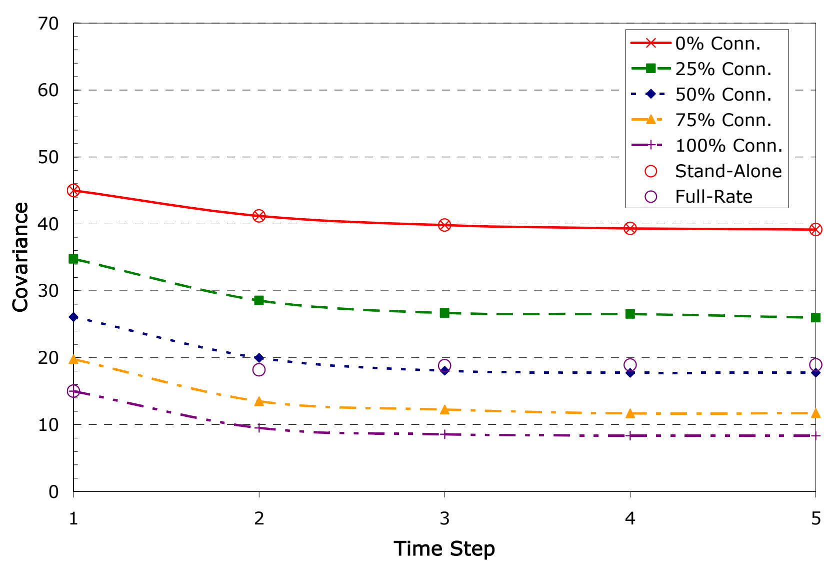

Figure 58: Naïve Communications Covariance Estimate versus Time for an Ad Hoc Communications Architecture under Steady-State Estimation Conditions.

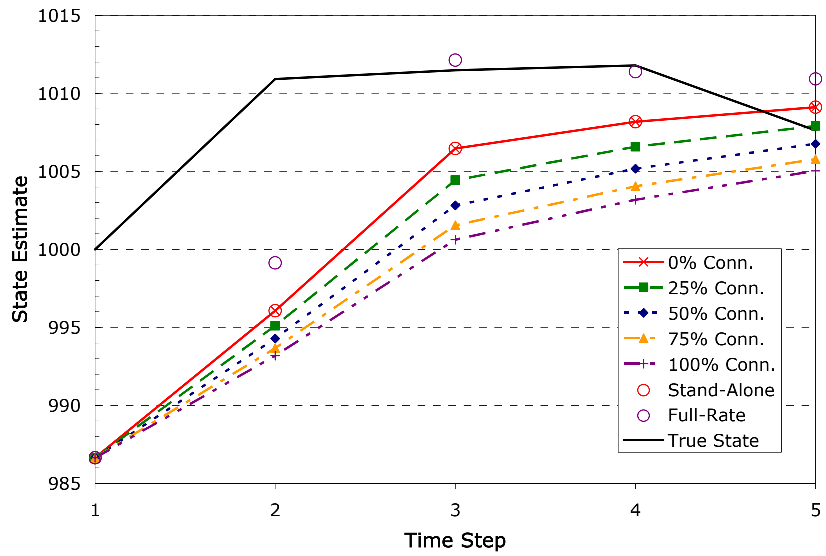

Figure 59: Naïve Communications State Estimate vs. Time for an Ad Hoc Communications Architecture under Steady-State Estimation Conditions.

significantly different from the Local Fusion Graph and optimal estimates for full-rate network connectivity. State estimation data actually shows decreasing estimation performance with increasing network connectivity for the naïve fusion method.

The reason for the poor performance of the naïve fusion method is that Equations (42) and (43) hold only if the fused estimates are mutually independent. With the presence of common process and observation noises, these assumptions do not hold [32]. Nonetheless, a comparative examination of communications requirements and estimation capabilities relative to the Local Fusion Graph is useful if only because of the popularity and simplicity of the naïve method.

Unlike the Local Fusion Graph, naïve fusion information exchanges contain no data relating to prior time steps. Thus the probability distribution of communication data rate for naïve fusion relative to network connectivity is given simply as a binomial distribution. The mean and maximum message sizes are thus given as $(n\text{-}1)\cdot\mu$ and n-1, respectively.

The mean message size relative to network connectivity is presented along with that of the Local Fusion Graph method in Figure 60. The data demonstrates that the average message size for naïve fusion in the ad hoc and broadcast networks is less than 30% of the Local Fusion Graph message size at 25% connectivity. The ratio increases to about a 75% at 75% connectivity because the Local Fusion Graph average message size quickly approaches the maximum while naïve data rate growth remains linear relative to network connectivity.

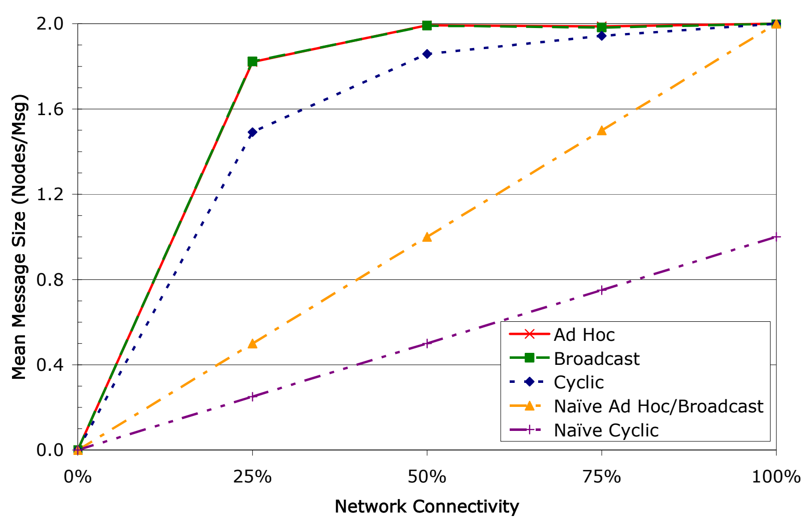

Figure 60: Network Connectivity vs. Average Message Size for Local Fusion Graph and Naïve Fusion Methods.

Given the accuracy and communications requirements together, it is observed that naïve fusion reduces communications requirements for distributed fusion networks relative to other approaches at the expense of estimation accuracy. The notional communications cost of the Local Fusion Graph, however, must be weighed against the increase in estimation performance. A comparison of the relationship between the two methods' communications requirements and average RMS error is shown in Figure 61 and Figure 62. Because the average message size increases with increasing network connectivity, Local Fusion Graph average RMS error trends with respect to message size are similar to RMS error trends relative to network connectivity. Naïve fusion average RMS error, however, is seen to increase in both the transient and steady-state scenarios with respect to average message size.

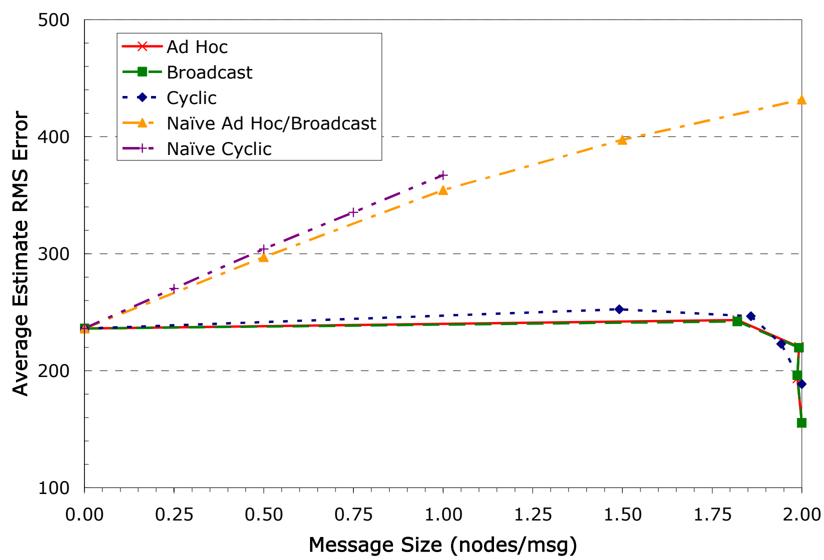

Figure 61: Distribution of Communications Load vs. Average Estimate RMS Error under Transient Estimation Conditions.

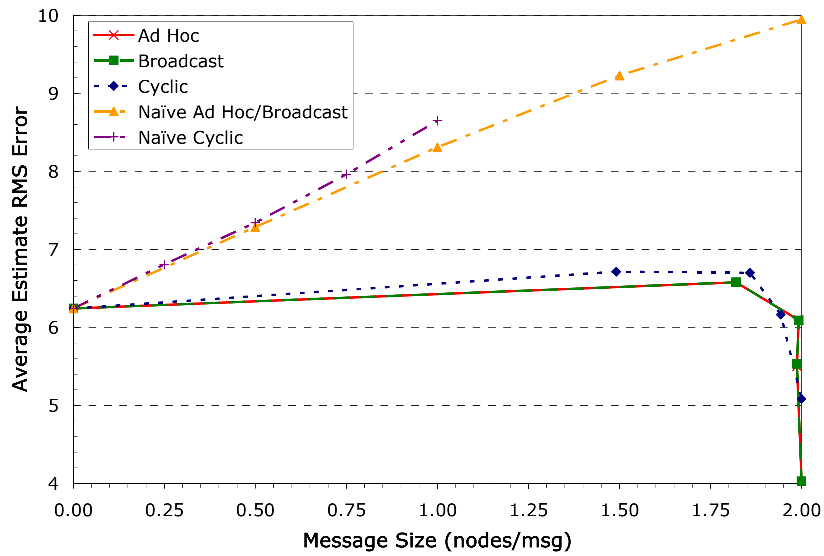

Figure 62: Distribution of Communications Load vs. Average Estimate RMS Error under Steady-State Estimation Conditions.

Under transient and steady-state estimation conditions modeled here, the RMS error reductions of the Local Fusion Graph are significant relative to any of the naïve fusion estimates. The primary reason for the difference in estimation performance is due to the presence of common estimates among the fusion agents. If no such dependencies were present, the results of the two methods would be comparable. In realistic implementations, however, such dependencies do exists. The added complexity of the Local Fusion Graph method is therefore worthy of consideration for the added estimation capabilities it provides.

3.4 Summary of Local Fusion Graph Simulation and Analysis

The results of the computer-based simulation of the Local Fusion Graph algorithm demonstrate the validity of the method for performing data fusion. The model implemented the estimation of a simple linear dynamic system in the presence of Gaussian state and measurement noises. The model's results are shown to precisely match those of known solutions at full network connectivity as well as the results for stand-alone estimation (e.g., no network connectivity). Further evidence of the method's validity is provided by its foundations in the Information Graph [36] and by the intuitive characteristics of the results produced for instances of non-deterministic network connectivity.

The performance of the Local Fusion Graph algorithm for non-deterministic network connectivity is demonstrated across a range of average network connectivity levels in transition and steady-state estimation conditions. In all cases, the average covariance is found to decrease in a non-linear manner with respect to increasing network connectivity. State estimates, however, demonstrate a more complex relationship with respect to network connectivity. If process noise is high relative to measurement noise, the state estimates are shown to increase a low non-zero average network connectivity levels relative to other connectivity levels. For lower ratios of process to measurement noise, however, the state estimates are shown to consistently decrease with respect to network connectivity.

The model also provided data for a first-order assessment of communication requirements of the Local Fusion Graph. While the assessment is limited in scope due to the dependence of actual bandwidth on protocol assumptions, the analysis demonstrates the trends in data rate relative to network connectivity. Specifically, average Local Fusion Graph communications requirements for network connectivity levels remain less than that of full-rate communications requirements. The peak data rate, however, will likely increase the communications requirements above that of full rate communications requirements. The comparison of communication requirements for the Local Fusion Graph and naïve fusion algorithms demonstrates the trades between communications loads and estimation performance.

The results of the communication analysis lead to some issues for further research. First, the statistical distributions of required information per message as a function of network connectivity could be used to develop a communications protocol that optimizes the amount of data exchanged between fusion agents. Further characterization of information needs relative to network size, connectivity, and mobility are required. A second yet related issue for further understanding is the characterization of missing data impacts on the Local Fusion Graph method. The analysis performed here considered only the case where all required data is received by each fusion agent. If a statistics-based protocol is used for inter-agent information exchange, agents may not always have complete data sets. The missing data will prevent the agents from identifying some common prior information sets, thus resulting in reduced estimation accuracy.

4 DERIVATION AND EVALUATION OF A STOCHASTIC FUSION FORMULATION

The aggregate behavior of communications networks are characterized by a number of stochastic phenomena. Network delay models are largely based upon queuing theory [11], and wireless channel characteristics are modeled by various random process models [7-9]. Since distributed information fusion is a function of communications capabilities, distributed fusion formulations should likewise contain those stochastic characteristics.

Chapter 2 develops an algorithm that enables the implementation and modeling of distributed fusion networks independent of communications characteristics. While the technique enables distributed fusion in the presence of ad hoc and stochastic communications characteristics, it does not produce an analytical formulation that is easily applied to estimation performance predictions. Predictions and analysis of average performance expectations such as those shown in Chapter 3 require non-trivial computer code development and large numbers of simulations. Therefore, a more compact or analytical solution is desired that reduces the complexity and computational requirements of simulation models and analyses.

For that purpose, this chapter develops a general formulation that captures the average behavior of distributed fusion in communications networks with arbitrary connectivity and stochastic link characteristics. The chapter further explores the use of some basic assumptions to produce a simplified stochastic formulation of distributed fusion in ad hoc networks. The simplified formulation is then developed into a computer-based model and the results are assessed relative to their consistency with the results found in Chapter 3. The assessments are accompanied by discussions for furthering the development and scope of the stochastic formulation.

4.1 Derivation of the Stochastic Fusion Formulation

Communications between fusion agents is characterized by transmit and receive probabilities as well as message delay probabilities. The effect of these stochastic characteristics was demonstrated in the example used for the Local Fusion Graph derivation. Combining the three local fusion graphs into a composite or centralized view is shown in Figure 63.

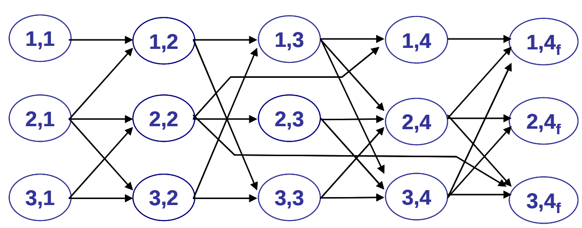

Figure 63: Example Composite Fusion Graph.

To conduct the derivation of a stochastic fusion formulation, an arbitrary fusion graph such as the one in Figure 63 is supposed. For simplicity in notation, each of the J

nodes in the fusion graph is given an index value j such that $1<=j<=J$. Similarly, the node representing the fusion event in question is designated by the index i.

As shown, data is transferred between nodes in the local fusion graph based on the probability of transmission, reception, and delay. To simplify the derivation of a stochastic fusion formulation without losing generality of its application, the aggregate effect of the various stochastic characteristics is captured in a single link probability factor. The average probability that data will be successfully transferred from node j to node i is given as μ_{ij}. Thus the average result for fusing information from node j at time k_j into node i at time k_i with some average probability μ_{ij} is given quite simply as:

$$\Phi\left(I_i(k_i)\right) = \mu_{ij}\left\{\Phi\left(I_i(k_i \mid k_i)\right) \cup \Phi\left(I_j(k_i \mid k_j)\right)\right\} \tag{44}$$

The overall average result at node j, however, must consider all possible outcomes. Each outcome includes a set S of nodes that transfer data to node i, and a set $\neg S$ of nodes that do not transfer data directly to node i which will be designated by. Assuming that all possible events are independent, the joint probability of data exchange events is given as:

$$\mu_S = \prod_{\substack{j=1 \\ j \in S}}^{J} \mu_{ij} \quad , j \neq i \tag{45}$$

Similarly, the joint probability of the events with no data exchanges is given as:

$$\mu_{\neg S} = \prod_{\substack{j=1 \\ j\in\neg S}}^{J} \left(1-\mu_{ij}\right) \quad ,j \neq i \tag{46}$$

For M possible outcomes, each defined by a unique fusion graph, the expected result at a given node across all possible outcomes is thus given as:

$$\Phi\left(I_i(k_i)\right) = \sum_{m=1}^{M} \mu_{S_m} \mu_{\neg S_m} \left[\left(\bigcup_{\substack{j=1 \\ j\in S_m}}^{J} \Phi\left(I_j(k_i \mid k_j)\right) \right) \cup \Phi\left(I_i(k_i \mid k_i)\right) \right] \quad ,j \neq i \tag{47}$$

Applying Equations (47) to the fusion equations from (2) and (3) results in the following stochastic fusion equations:

$$\hat{\mathbf{x}}_i(k_i) = \sum_{m=1}^{M} \mu_{S_m} \mu_{\neg S_m} \mathbf{P}_m(k_i) \begin{bmatrix} \mathbf{P}_i^{-1}(k_i \mid k_i)\hat{\mathbf{x}}_i(k_i \mid k_i) + \sum\limits_{\substack{j=1 \\ j\in S_m}}^{J} \mathbf{P}_j^{-1}(k_i \mid k_j)\hat{\mathbf{x}}_j(k_i \mid k_j) - \\ \overline{\mathbf{P}}_i^{-1}(k_i)\overline{\mathbf{x}}_i(k_i) \end{bmatrix} \tag{48}$$

$$\mathbf{P}_i(k_i) = \sum_{m=1}^{M} \mathbf{P}_m(k_i) = \sum_{m=1}^{M} \mu_{S_m} \mu_{\neg S_m} \left[\mathbf{P}_i^{-1}(k_i \mid k_i) + \sum_{\substack{j=1 \\ j\in S_m}}^{J} \mathbf{P}_j^{-1}(k_i \mid k_j) - \overline{\mathbf{P}}_i^{-1}(k_i) \right]^{-1} \tag{49}$$

While this formulation appears simple, its practical implementation can become very complex if the probability factors for each node pair are different. The solution becomes even more complex if delays are present. If the probability all possible node pairs is non-zero, then a fusion graph containing n fusion agents and a history of k_i time steps must evaluate as many as $2^{k_i(n-1)}$ possible outcomes for each node in the fusion graph. Furthermore, Equations (48) and (49) do not enumerate the common prior information that must be found for each possible combination of fusion events. Each set *S* that is fused has multiple possible prior histories that must be considered. Developing a stochastic representation of the common prior information for these histories cannot be represented in any simple formulation. Instead, methodical searches using computer-based algorithms may be the only means for valuating the common prior information.

The extensive efforts required for developing a stochastic representation of the common prior information is not pursued here due to the scope of the current thesis work. Instead, some simplifying assumptions are applied to Equation (47) in the next section to develop an approximation for distributed fusion in ad hoc networks. The resulting formula is implemented in a computer-based model and the results are presented. Assessments are provided relative to the Local Fusion Graph results for comparisons of the simplified stochastic formulation's estimation capabilities.

4.2 Derivation of a Simplified Stochastic Formulation for Distributed Fusion in Ad Hoc Networks

The derivation of the general stochastic formulation in the previous section allows for each node pair in the fusion graph to have a unique link probability value. By properly selecting the link probabilities, any type of communications pattern can be represented. In essence, the communications event generator used in the Local Fusion Graph simulation model applies this approach. It allows the user to establish templates for communications patterns where some links are disabled by setting μ_{ij}=0 and others are implemented with some non-zero link probability. Thus applying similar assumptions to the general stochastic formulation in Equation (47) should yield the same results as the Local Fusion Graph model.

Three basic assumptions were made for the data produced by the Local Fusion Graph model:

1) Fusion agents utilize equivalent state and measurements models, including initial estimates and covariance;
2) The connectivity probabilities between agents are equivalent for all agent pairs;
3) Messages are sent and received within the same time step.

The assumptions establish uniformity across the fusion agents and allow the effects of average connectivity to be the primary factor in the resulting data trends. With the second and third assumptions, the link probability factors for each combination represented in Equation (47) can be simplified to a binomial. The joint probability of *m* data sets received by node *i* for fusion is then given by:

$$\mu_S \mu_{\neg S} = \beta(m; n-1, \mu_S) = \binom{n-1}{m} \mu_S^m (1-\mu_S)^{(n-1)-m} \tag{50}$$

The uniformity assumptions also result in all agents having the same value at a given time step. Thus, removing the subscripts from Equation (47) and applying Equation (50) yields the simplified stochastic formulation for ad hoc networks:

$$\Phi\left(I_i(k_i)\right) = \sum_{m=0}^{n-1} \binom{n-1}{m} \mu_S^m \left(1-\mu_S\right)^{(n-1)-m} \left(\bigcup_{j=1}^{m} \Phi\left(I(k \mid k)\right) \right) \tag{51}$$

While the simplifying assumptions reduce the order of the common prior information in Equation (51), they do not provide a simplified statistical formulation in determining the value of the common prior information. Therefore, an approximate valuation of the prior information must be made. Given the very similar results of ad hoc and broadcast network models in the Local Fusion Graph analysis, the common data term for a fully-connected fusion network operating in broadcast mode is used:

$$\Phi\left(\bar{I}_i(k_i)\right) = \left(m-1\right) \cdot \Phi\left(I_i(k_i \mid k_i - 1)\right) \tag{52}$$

Applying Equations (51) and (52) to the fusion equations from (2) and (3) results in the following simplified stochastic fusion equations:

$$\hat{\mathbf{x}}(k) = \mathbf{P}(k) \sum_{m=0}^{n-1} \beta(m; n-1, \mu_S) \left[(m+1)\mathbf{P}^{-1}(k \mid k)\hat{\mathbf{x}}(k \mid k) - m \cdot \mathbf{P}^{-1}(k \mid k-1)\hat{\mathbf{x}}(k \mid k-1) \right] \tag{53}$$

$$\mathbf{P}(k) = \sum_{m=0}^{n-1} \left\{ \beta(m; n-1, \mu_S) \left[(m+1)\mathbf{P}^{-1}(k \mid k) - m \cdot \mathbf{P}^{-1}(k \mid k-1) \right]^{-1} \right\} \tag{54}$$

where:

$$\beta(m; n-1, \mu_S) = \binom{n-1}{m} {\mu_S}^m \left(1-\mu_S\right)^{(n-1)-m}$$

The simplified stochastic fusion formulation presented by Equations (53) and (54) are thus proposed as distributed estimation performance approximations for ad hoc and broadcast networks with non-deterministic connectivity. The following section describes the development of these formulations into a simulation model and provides an analysis of their accuracy.

4.3 Performance Analysis of the Simplified Stochastic Ad Hoc Fusion Formulation

As with the Local Fusion Graph method, the stochastic fusion formulation was developed into a computer-based model to assess its performance. The stochastic model, shown in Figure 64, is implemented in MATLAB® and uses the same state and measurements model as the Local Fusion Graph model. The stochastic model, however, uses a connectivity matrix containing the connectivity probabilities (e.g., μ_{ij}) between each agent for each time step in the simulation rather than the event-based communications generator used in the Local Fusion Graph model. The model calculates the state and covariance estimates using the communications matrix and Equations (53)

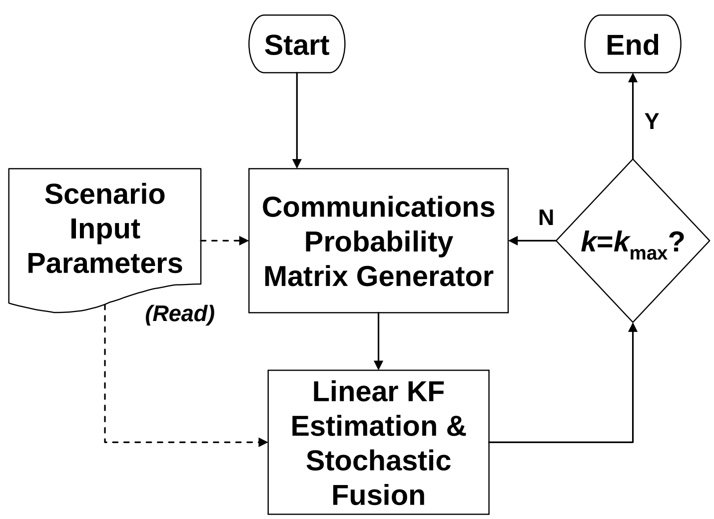

Figure 64: Stochastic Fusion Simulation Model Block Diagram.

and (54) at each time step. Because the model uses a stochastic representation of communications events rather than event-based one, multiple Monte Carlo runs not required for a given state and measurements model.

In the cases presented here, the state and measurements models used in Chapter 3 are also used here to allow for a direct comparison between the two models. Table 4 presents the values used in the simulations. Of particular note is the fact that the results obtained from 1000 Monte Carlos of the Local Fusion Graph model that require a total run time of approximately 12,000 seconds can now be approximated with a single run of the stochastic model of less than 2 seconds. The reduced simulation time is accompanied by a significant reduction in coding complexity.

Table 4: Parameter Values for Stochastic Fusion Simulations.

Parameter	**Value**	**Parameter**	**Value**
B, F,G,H	1	$\mathbf{x}(0)$	1000+N(0,**Q**)
u	0	Monte Carlos	1
v, Q	N(0,**Q**), 25	Fusion Agents	3
w, R	N(0,**R**), 100	k_0, k_{max}	1, 5
$\mathbf{P}_{trans}(k_0)$, $\hat{\mathbf{x}}_{trans}(k_0)$	**R**, N(0,**R**)	Message Delay	0
$\mathbf{P}_{ss}(k_0)$, $\hat{\mathbf{x}}_{ss}(k_0)$	45, $\mathbf{x}(0)$+N(0,45)	$\mu_{ij}(k)$	0, 0.25, 0.5, 0.75, 1

As with the Local Fusion Graph analyses, the analysis of results from the simplified stochastic fusion approximation include covariance and state estimation for a number of network connectivity levels. The comparisons of the two models are performed using an ad hoc communications architecture under transient and steady state conditions. As discussed in the derivation of the simplified stochastic formulation, the formulation's results hold for both broadcast and ad hoc network architectures due to their statistically similar results.

Two types of comparisons are made between the models. First, comparisons of predicted results are made to assess the magnitude of any differences. These comparisons provide insight into how well the simplified stochastic fusion method replicates the estimation performance of the Local Fusion Graph approach. The second comparison type assesses how well the stochastic fusion graph imitates the trends of the Local Fusion Graph relative to network connectivity.

Comparisons of the covariance estimations for the Local Fusion Graph and simplified stochastic algorithm in an ad hoc communications architecture under transient and steady-state estimation conditions is shown in Figure 65 and Figure 66, respectively. The data demonstrate significant agreement between the two models under both estimation conditions. At k=1, the simplified stochastic model and average Local Fusion Graph results are precisely the same because no common prior information exists among the fusion agents. Thus no difference will exist in the simulation data provided both models utilize the same initial conditions. The difference between them is shown to increase in the first few time steps due to the approximation made for the common prior information. With the exception of the 25% connectivity case, the stochastic approximations and Local Fusion Graph estimates are then seen to converge with time.

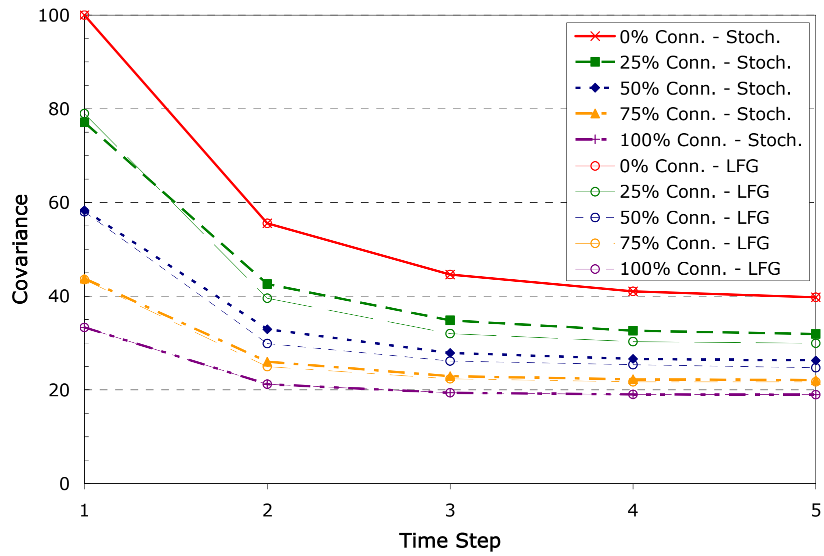

Figure 65: Comparison of Ad Hoc Covariance vs. Time for the Local Fusion Graph and Simplified Stochastic Fusion Methods under Transient Estimation Conditions.

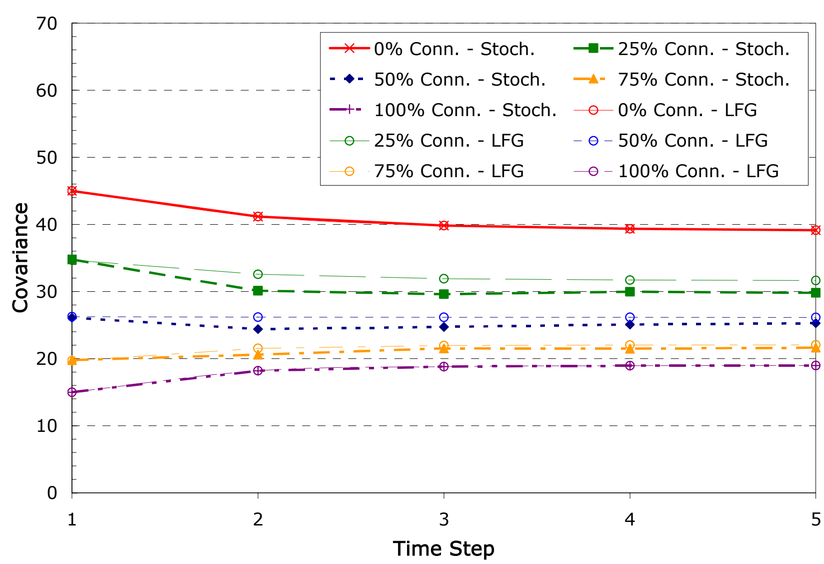

Figure 66: Comparison of Ad Hoc Covariance vs. Time for the Local Fusion Graph and Simplified Stochastic Fusion Methods under Steady-State Estimation Conditions.

Differences in results can also be seen in Figure 67, which compares the trends in covariance averaged across time versus connectivity under transient and steady-state estimation conditions, respectively. Again the data demonstrate similar trends and magnitudes of covariance relative to network connectivity. Under the random connectivity cases, the simplified stochastic formulation covariance approximations are shown to be more accurate at high network connectivity levels. Figure 68 presents the covariance differences of the two methods at each connectivity level normalized by the initial covariance. The data demonstrates that the maximum difference between the two covariance results occurs at 25% connectivity.

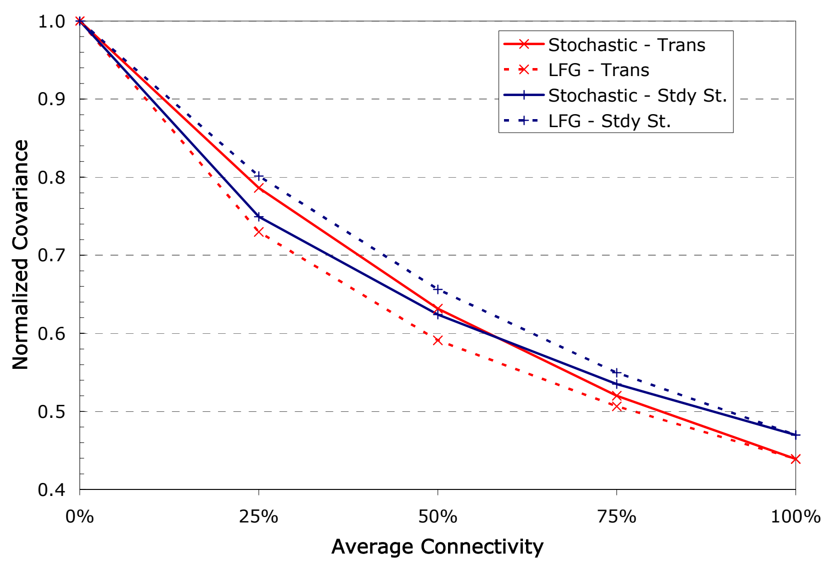

Figure 67: Normalized Average Covariance vs. Connectivity under Transient and Steady-State Estimation Conditions.

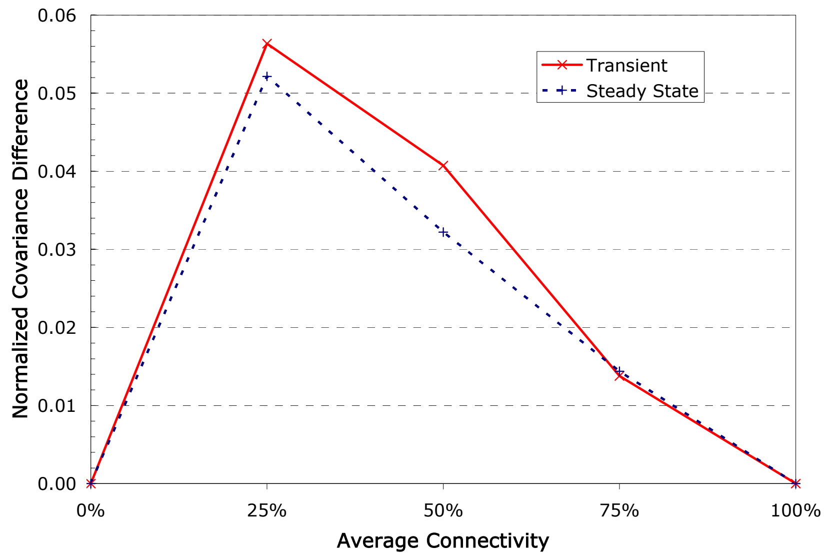

Figure 68: Normalized Covariance Estimation Difference vs. Connectivity under Transient and Steady-State Estimation Conditions.

As with the covariance approximations, the simplified stochastic formulation closely approximates the state estimates at each of the time steps as shown in Figure 69 and Figure 70. A comparison of the normalized RMS errors produced by the two methods relative to network connectivity, however, demonstrates a difference in trends as shown in Figure 71. While the Local Fusion Graph produces an initial increase in RMS error at 25% connectivity for both cases, the simplified stochastic formulation predicts a sizeable decrease. The resulting difference in RMS error is plotted at each connectivity level in Figure 72.

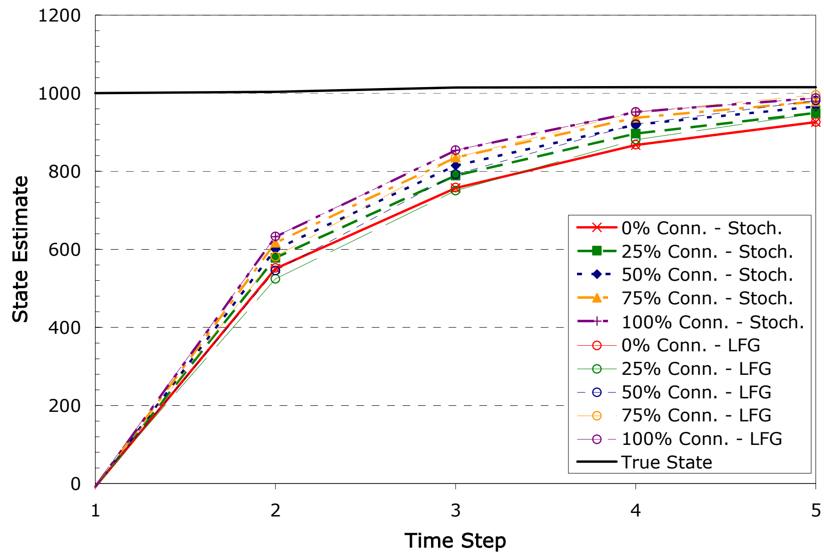

Figure 69: Comparison of Ad Hoc State vs. Time for the Local Fusion Graph and Simplified Stochastic Fusion Methods under Transient Estimation Conditions.

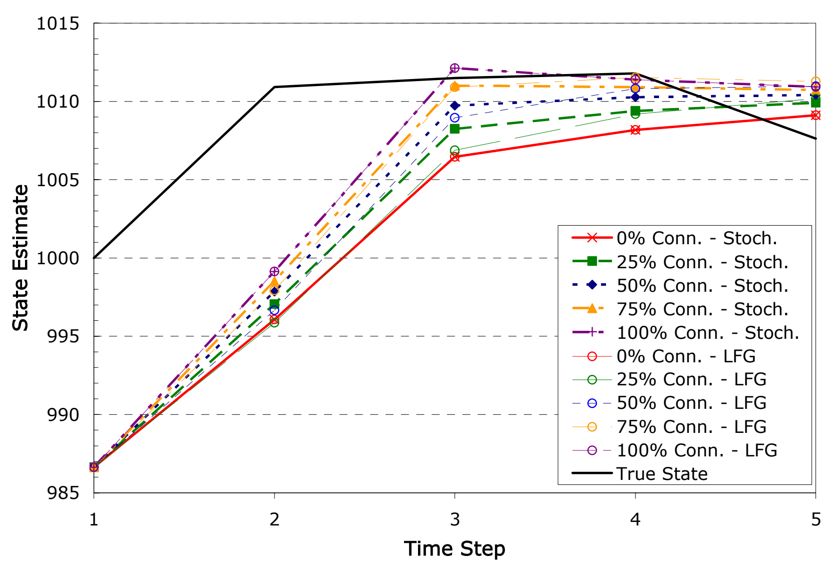

Figure 70: Comparison of Ad Hoc State vs. Time for the Local Fusion Graph and Simplified Stochastic Fusion Methods under Steady-State Estimation Conditions.

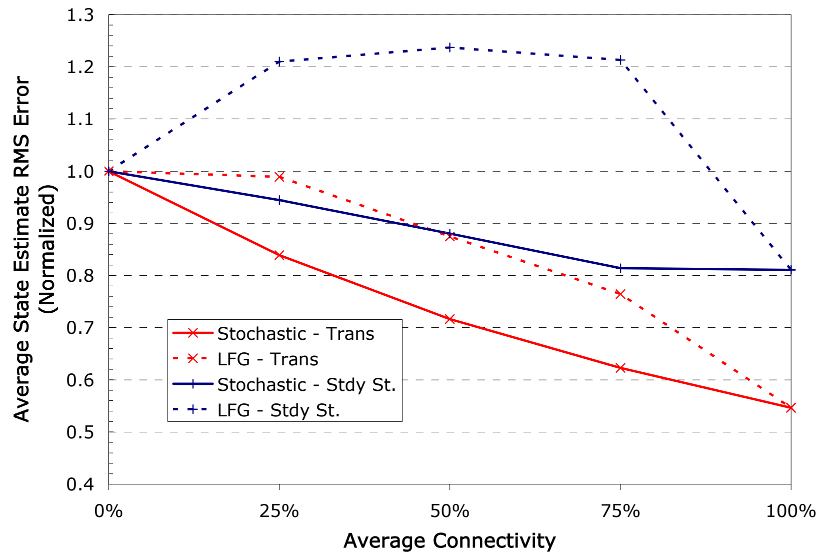

Figure 71: Average State RMS Error Estimates vs. Network Connectivity under Transient and Steady-State Estimation Conditions.

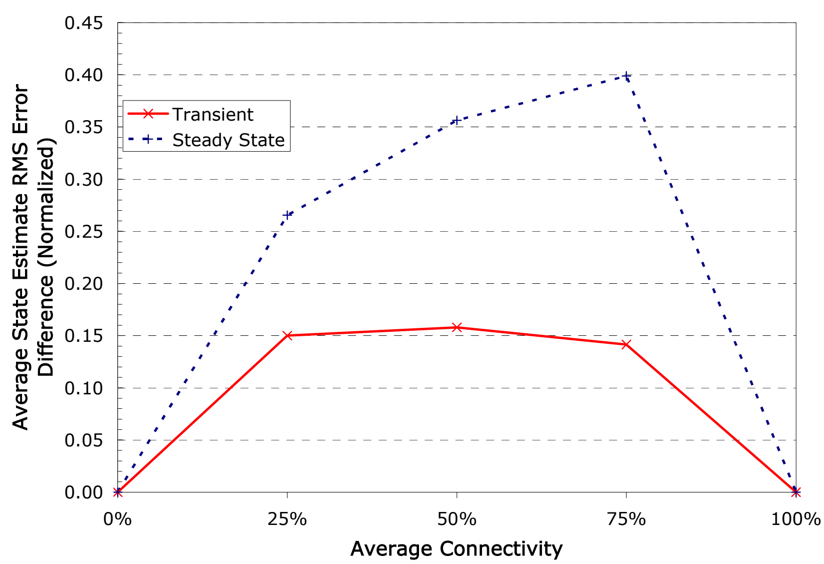

Figure 72: Differences in Average State RMS Error Estimates vs. Network Connectivity under Transient and Steady-State Estimation Conditions.

The reason for these differences can be understood in light of the discussion presented in Chapter 3 regarding the increased average RMS error at low network connectivity levels. The results of Local Fusion Graph stem from impacts to the filter gain calculation due to inconsistent network connectivity experienced in individual event-based simulations. The stochastic formulation, however, utilizes an average connectivity at each time step rather than the connectivity fluctuations that produce the increased RMS errors in the Local Fusion Graph results.

Regardless of the difference in estimation trends, the data presented in the comparative analysis demonstrates that the simplified stochastic formulation provides a close approximation of the Local Fusion Graph estimates for ad hoc networks provided the simplifying assumptions are applicable. Comparisons of covariance and state estimates show only small variations in the magnitude of their differences, with the maximum occurring at low network connectivity levels. As evidenced throughout the Local Fusion Graph analysis, the results for ad hoc and broadcast networks are nearly identical. Thus the simplified stochastic method is applicable for estimation approximations for broadcast networks as well.

4.4 Summary of Stochastic Fusion Formulation and Analysis

A stochastic fusion formulation is proposed that incorporates average network connectivity and message delay into the fusion equations. The resulting equations are the probability-weighted sums of all possible outcomes at each time step and are given in Equations (48) and (49) as:

$$\hat{\mathbf{x}}_i(k_i) = \sum_{m=1}^{M} \mu_{S_m} \mu_{\neg S_m} \mathbf{P}_m(k_i) \left[\begin{array}{l} \mathbf{P}_i^{-1}(k_i \mid k_i)\hat{\mathbf{x}}_i(k_i \mid k_i) + \sum_{\substack{j=1 \\ j \in S_m}}^{J} \mathbf{P}_j^{-1}(k_i \mid k_j)\hat{\mathbf{x}}_j(k_i \mid k_j) - \\ \overline{\mathbf{P}}_i^{-1}(k_i)\overline{\mathbf{x}}_i(k_i) \end{array} \right]$$

$$\mathbf{P}_i(k_i) = \sum_{m=1}^{M} \mathbf{P}_m(k_i) = \sum_{m=1}^{M} \mu_{S_m} \mu_{\neg S_m} \left[\mathbf{P}_i^{-1}(k_i \mid k_i) + \sum_{\substack{j=1 \\ j \in S_m}}^{J} \mathbf{P}_j^{-1}(k_i \mid k_j) - \overline{\mathbf{P}}_i^{-1}(k_i) \right]^{-1}$$

Implementations therefore comprise combinatory sets of current and common prior estimates that grow exponentially with time and the number of fusion agents in the network. Furthermore, identification of the prior information in a stochastic manner is not easily represented in any simple probability formulation since the occurrence of common prior information is a function of the specific histories. While these equations enable analyses of estimation performance in networks with stochastic characteristics, they require a computer-based implementation for practical assessments.

To reduce the computational complexity, a simplified stochastic formulation for ad hoc networks was derived and implemented in a computer-based model. The simplified formulation is achieved by assuming no message delays, homogenous estimation capabilities for each agent, and equivalent message delivery probabilities for all agent pairs. The resulting stochastic fusion formulations are given in Equations (53) and (54) as:

$$\hat{\mathbf{x}}(k) = \mathbf{P}(k) \sum_{m=0}^{n-1} \beta(m; n-1, \mu_S) \left[(m+1)\mathbf{P}^{-1}(k \mid k)\hat{\mathbf{x}}(k \mid k) - m \cdot \mathbf{P}^{-1}(k \mid k-1)\hat{\mathbf{x}}(k \mid k-1) \right]$$

$$\mathbf{P}(k) = \sum_{m=0}^{n-1} \left\{ \beta(m; n-1, \mu_S) \left[(m+1)\mathbf{P}^{-1}(k \mid k) - m \cdot \mathbf{P}^{-1}(k \mid k-1) \right]^{-1} \right\}$$

where:

$$\beta(m; n-1, \mu_S) = \binom{n-1}{m} \mu_S{}^m (1-\mu_S)^{(n-1)-m}$$

Comparisons of the simplified formulation results with those of the Local Fusion Graph demonstrate the ability to closely approximate the average distributed estimation performance in ad hoc and broadcast networks. One notable exception is that the stochastic method does not replicate the increase in average RMS error that is encountered at low network connectivity levels for some scenarios. In addition to the good approximation capabilities, the stochastic method is much simpler to code into a computer-based model than the Local Fusion Graph method, and the processing time is reduced by a factor of 6000.

5 SUMMARY AND RECOMMENDATIONS

This thesis proposes two complimentary approaches as general solutions to distributed information fusion in networks with ad hoc connectivity and stochastic link characteristics. The Local Fusion Graph method is a means for implementing information fusion in distributed fusion agents without requiring a priori knowledge of network membership, connectivity, or communications patterns. The stochastic fusion formulation incorporates average message delay and delivery probabilities into the fusion equations and permits analyses of expected estimation performance in fusion networks having non-deterministic connectivity. Both methods are derived and the resulting formulations are implemented in computer-based models. The results of the models are analyzed to assess the validity of the techniques and gain insight into distributed estimation capabilities under non-ideal network connectivity conditions.

5.1 Summary of Local Fusion Graph Findings

The Local Fusion Graph methodology is proposed as a general solution for distributed fusion in networks with non-deterministic communications connectivity such as mobile ad hoc networks. The method is grounded in set-theoretic derivations of information fusion. More specifically, the methodology is based in large part on the logic of the Information Graph proposed in [36]. Where the Information Graph is used to provide a centralized view of distributed fusion events from an omniscient perspective, the Local Fusion Graph approach allows each fusion agent to build and maintain its own localized graph. The local fusion graph provides all information required for calculating the locally-fused estimate at each agent. The technique requires no a priori knowledge of network membership, connectivity, or communications patterns and is applicable to any arbitrary distributed fusion network regardless of communications patterns or message delays.

The validity and performance capabilities of the Local Fusion Graph are demonstrated through mathematical and analytical assessments. The method's basis in the set-theoretic principles and utilization of the Information Graph logic as a foundation lend to its credibility. It is shown that the Local Fusion Graph can be used to produce fusion equations consistent with known solutions. Furthermore, the results of a computer-based simulation demonstrate consistency with results from the known formulations.

The benefits of the Local Fusion Graph are apparent in the analysis of the computer-based simulation results. Where current methods are only accurate in networks with deterministic connectivity, the Local Fusion Graph method can perform under any communications networking conditions. The analysis characterizes the method's performance capabilities for conducting distributed estimation under transient and steady-state estimation conditions. It also demonstrated decay in the accuracy of distributed estimations based on the Kalman Filter at low network connectivity levels. A first-order analysis of Local Fusion Graph communications requirements shows that the average data rate for the local fusion graph may be comparable with that of a fully-connected network, but peak rates can exceed the average rates by an order of magnitude in some cases. Overall, the Local Fusion Graph provides significant benefit for information fusion in uncertain networking environments. The resulting algorithms are able to be implemented in distributed fusion agents without a priori knowledge of communications infrastructures, and provide the ability to perform fusion operations in challenging network environments.

5.2 Summary of Stochastic Fusion Formulation Findings

Given the results of the Local Fusion Graph, an analytical approach to predicting and assessing distributed estimation in networks with stochastic connectivity was explored. The examination sought to produce a relatively simple representation of distributed fusion premised upon the stochastic nature of wireless communications. The fundamental factors driving the uncertainty in distributed fusion networks are encapsulated in message delivery and delay probabilities. A stochastic formulation is proposed that captures the probabilistic characteristics, but a compact formulation of common prior information could not be represented in a stochastic manner. The occurrence of common prior information is found to be a combinatoric function of the specific histories, and producing them in a complete and general manner is beyond the scope of this effort.

A simplified variant of the general stochastic fusion formulation was developed to explore approximate results for ad hoc networks. The simplified form is derived by assuming homogenous network connectivity, message delays, and state models. Additionally, the common prior information is approximated using the common priors term of the broadcast formulation. The resulting simplified stochastic formulation is compact and greatly reduces the complexity of distributed estimation analysis in ad hoc networks.

A computer-based model was developed to assess the simplified stochastic formulation's ability to approximate Local Fusion Graph results. Analysis demonstrated that the formulation results closely approximate the averaged results of the Local Fusion Graph with few exceptions. Additionally, coding the stochastic method into a computer-based model is much simpler than the Local Fusion Graph model, and the processing time is reduced by three orders of magnitude. Thus the simplified stochastic fusion formulation has the ability to enable relatively simple analyses of estimation capabilities and trends in ad hoc networks.

5.3 Recommendations for Future Research

While the derivations, analysis, and results of this study captured a wide range of issues related to distributed estimation in networks with stochastic connectivity, a number of open questions for further research still remain. The analyses performed here established a basic understanding of the algorithm's performance and related issues. Extending the findings into general characteristics and principles requires a more thorough treatment than is practical in this effort.

The first area to explore is to determine if the Local Fusion Graph preserves optimality of estimation models with respect to the underlying network connectivity. Whether the method produces locally-optimal results given the information available to each fusion agent may depend on the application that utilizes the algorithms. Estimation formulations that are optimal under fully-connected networks may not be optimal under varying network connectivity conditions. Chapter 3 demonstrated that RMS errors can increase at low non-zero connectivity levels relative to stand-alone estimation. Thus dynamic network connectivity conditions may preclude optimality for some estimation techniques.

The second area for research is the establishment of an effort to further characterize Local Fusion Graph performance. The analysis conducted here was necessarily focused in scope and purpose. It was conducted primarily for validating the Local Fusion Graph algorithms and providing a first-order assessment of the trends regarding estimation performance relative to network connectivity. Aspects to be investigated include scalability of performance relative to the number of fusion agents, accuracy of the method in the presence of missing data, and stability of the estimates under a range of network conditions.

Future efforts should also investigate efficient communications and computational aspects of the Local Fusion Graph method such as the design of efficient data encoding techniques and communications protocols. Because the method utilizes a graph to organize the data, graph-based coding theories such as those used for bioinformatics [52] may lead to efficient searches for common prior data sets as well as efficient message encoding. Another technique for message encoding is to utilize a probabilistic approach similar digital message encoding, where the level of information redundancy can be varied based on the desired probability of success.

A fourth avenue for investigation is the exploration of alternative estimation techniques under the Local Fusion Graph construct. Among them is whether the method can be approached from a Bayesian Network perspective. The combination of the Information Graph and Bayesian Network concepts for identifying communications and measurements dependencies in [39] should be extensible to the Local Fusion Graph. With an initial inspection, the entire Local Fusion Graph may lend itself to be viewed as a Bayesian Network [53]. Thus it may be possible to construct the fusion algorithms in a manner that avoids the lengthy process of recursively searching for and removing common data elements. Resulting methods would need to be assessed in the context of enabling simplified analysis as well as implementation in distributed fusion networks.

A final area for further investigation of the Local Fusion Graph approach is to determine the applicability and impact of the method on higher level fusion. The method is inherently designed to manage information for each local fusion agent and should therefore be applicable to managing information such as evidence and beliefs for distributed reasoning approaches [53]. The Local Fusion Graph may provide increased abilities to conduct distributed reasoning in ad hoc networks.

In addition to the Local Fusion Graph, the stochastic formulation should be pursued and developed further. An effort should be taken up to compactly and analytically characterize common prior data for the general stochastic fusion formulation. If that effort produces simplified results relative to the initial efforts of this study, then the ability to accurately predict estimation performance in any networking environment will be greatly simplified.

APPENDIX A: PRINCIPLES FOR DETERMINING THE TIME-UPDATED VALUE OF INFORMATION

As presented throughout the main sections of this document, effective distributed information fusion requires that common information elements be identified and removed before the sets are fused. The common information elements for the class of estimation systems addressed here are estimations that were determined at some prior time. While the estimates were of a certain value at the time of their calculation, that estimate may have a different value at subsequent times. Therefore, the removal of the common prior estimates must utilize the present (or "time-updated") value of the common prior estimate.

A search for the mechanisms and principles for managing and calculating the value of prior estimates produced little insight. The sources used throughout this document discuss the need to determine the present value of common prior estimates, but no methods or discussions of algorithms are presented. Given these facts, this appendix provides a derivation of the time-update formulations used in the distributed fusion graphs and models. While the following discourses are not designed as formal proofs, they adhere to mathematical and logical theories to provide evidence for divining appropriate principles and methods.

The fusion of information sets containing data dependencies produce common prior sets that can be categorized into the following three general classes:

1) Single common prior information set
2) Multiple common prior information sets with common time references
3) Multiple common prior information sets with dissimilar time references

Valuation of single common prior sets is rather trivial as they are easily found through recursive applications of prediction operations of the estimation method. The discussion of that case, however, is used as a basis for understanding the other two cases and is necessarily included.

All three cases use a collection of information sets and associated transitions between them, such as that shown in Figure 73. The sets are grouped according to common reference frames (such as state or time) as indicated by the dashed lines. As the information from a set crosses these boundaries, its value transitions into one or more sets in the new reference frame. For derivation purposes, the equation governing the transitions from some set "X" into set "Y" across a boundary is given as:

$$I_X(Y) = f\left(I_Y, a_1 \ldots a_M\right) = \sum_{k=0}^{M} a_k I_Y^k \tag{55}$$

with arbitrary coefficients a_k. This equation is used as the basis for determining the appropriate approach in all three priors cases.

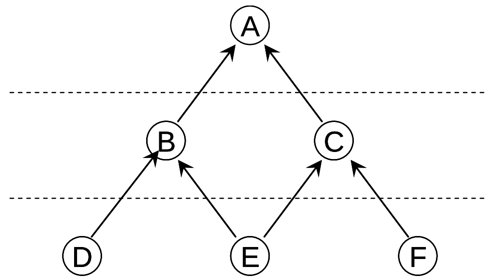

Figure 73: Network of Information Sets with a Single Common Prior.

Valuation of Single Common Priors

The valuation of single common priors can be determined by considering the network shown in Figure 73. The value of information set A (represented as I_A) is given by applying Equation (1) to fuse the current values of I_B and I_C as follows:

$$\begin{aligned} I_A &= I_A(B) \cup I_A(C) \\ &= I_A(B) + I_A(C) - \bar{I}_A \\ &= I_A(B) + I_A(C) - I_A(E) \end{aligned} \tag{56}$$

The values of $I_A(B)$ and $I_A(C)$ are found by Equation (55):

$$\begin{aligned} I_A(B) &= \sum_{k=0}^{M} a_k I_B^k \\ I_A(C) &= \sum_{k=0}^{M} a_k I_C^k \end{aligned} \tag{57}$$

Similarly, I_B and I_C are determined by $I_B(D) \cup I_B(E)$ and $I_C(E) \cup I_C(F)$ respectively. With independence between I_D, I_E, and I_F, Equation (57) then becomes:

$$
\begin{aligned}
I_A(B) &= \sum_{k=0}^{M} a_k \left(I_B(D) \cup I_B(E)\right)^k = \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_D^j \cup \sum_{j=0}^{M} a_j I_E^j \right)^k \\
I_A(C) &= \sum_{k=0}^{M} a_k \left(I_C(E) \cup I_C(F)\right)^k = \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_E^j \cup \sum_{j=0}^{M} a_j I_F^j \right)^k
\end{aligned}
\tag{58}
$$

From Equation (58), it can be seen that the common information is a function of the transformed value of I_E. For linear dynamic systems (M = 1) such as those addressed in this paper, the common information is easily formulated to be:

$$
\bar{I}_A = \sum_{k=0}^{1} a_k \left(\sum_{j=0}^{1} a_j I_E^j \right)^k = a_1 \left(a_1 I_E + a_0\right) + a_0 \tag{59}
$$

It is therefore shown that the value of common information to be removed from a fused data set is found by recursively transforming its value at the original reference frame to the reference frame of the fusion event. For KF-based estimation, Equation (59) can be shown as:

$$
\bar{\mathbf{x}}_A = \mathbf{F}\left(\mathbf{F}\hat{\mathbf{x}}_E + \mathbf{Bu} + \mathbf{Gv}\right) + \mathbf{Bu} + \mathbf{Gv} \tag{60}
$$

$$
\bar{\mathbf{P}}_A = \mathbf{F}\left(\mathbf{F}\mathbf{P}_E\mathbf{F}' + \mathbf{GQG}'\right)\mathbf{F}' + \mathbf{GQG}' \tag{61}
$$

Valuation of Multiple Common Priors with Common Time References

The approach used for single common prior set valuation is extended to fusion events with multiple common prior sets that contain the same time reference. Using the graph shown in Figure 74, the information sets I_F and I_G are the common sets for I_B, I_C, and I_D when fused into I_A. The question at hand is whether I_F and I_G are fused at the original reference frame and then transformed to the current reference frame, or if the transformation is performed on the two data sets before they are fused.

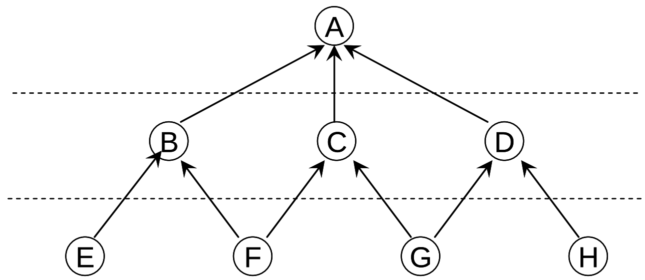

Figure 74: Network of Information Sets with Multiple Common Priors at the Same Reference Frame.

The two update options can be represented respectively as follows:

$$\begin{aligned} I_A &= I_A(B) + I_A(C) + I_A(D) - \left(I_F(F) \cup I_G(G)\right)_A \\ I_A &= I_A(B) + I_A(C) + I_A(D) - \left(I_A(F) \cup I_A(G)\right) \end{aligned} \tag{62}$$

The consequences of the different prior formulations in the context of the transformations used here are given as follows:

$$\begin{aligned} \left(I_F \cup I_G\right)_A &= \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_F^j \cup \sum_{j=0}^{M} a_j I_G^j \right)^k \\ I_A(F) \cup I_A(G) &= \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_F^j \right)^k \cup \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_G^j \right)^k \end{aligned} \tag{63}$$

These two formulations are not generally equivalent. For example, the equations in (63) applied to linear systems become:

$$\begin{aligned}
&(I_F \cup I_G)_A = a_1\left[(a_1 I_F + a_0) \cup a_1 (a_1 I_G + a_0)\right] + a_0 \\
&I_A(F) \cup I_A(G) = \left[a_1 (a_1 I_F + a_0) + a_0\right] \cup \left[a_1 (a_1 I_F + a_0) + a_0\right]
\end{aligned} \tag{64}$$

To determine the appropriate valuation of I_F and I_G in the fused sets $I_A(B)$, $I_A(C)$, and $I_A(D)$, the fusion of the three sets at I_A are given as:

$$I_A = I_A(B) \cup I_A(C) \cup I_A(D) = \left[I_A(B) \cup I_A(D)\right] \cup I_A(C) \tag{65}$$

From Figure 74 it is given that $I_A(B)$ and $I_A(D)$ are independent, and the formulation becomes:

$$I_A = I_A(B) + I_A(C) + I_A(D) - \left[I_A(B) + I_A(D)\right] \cap I_A(C) \tag{66}$$

Further investigation of the diagram shows that I_C is composed entirely of the common elements I_F and I_G. Thus the fusion equation given by (66) becomes:

$$\begin{aligned}
I_A &= I_A(B) + I_A(D) + I_A(C) - \left[I_A(B) \cap I_A(C) + I_A(D) \cap I_A(C)\right] \\
&= I_A(B) + I_A(D) + I_A(C) - \left[I_A(F) + I_A(G)\right] \\
&= I_A(B) + I_A(D) + I_A(C) - \left[I_A(C)\right] \\
&= I_A(B) + I_A(D)
\end{aligned} \tag{67}$$

The implication is that and the common information is equivalent to $I_A(C)$:

$$I_A(C) = \sum_{k=0}^{M} a_k \left(I_C(F) \cup I_C(G)\right)^k = \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_F^j \cup \sum_{j=0}^{M} a_j I_G^j\right)^k \tag{68}$$

Noting that Equation (68) is equivalent to the first formulation from (63), the method for valuating the multiple common prior sets is given as:

$$\left(I_F \cup I_G\right)_A = \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_F^j \cup \sum_{j=0}^{M} a_j I_G^j \right)^k \tag{69}$$

Namely, the valuation of multiple common prior sets with common reference frames is found by first fusing the information sets at their originating reference frames before translating to the reference frame of the current fusion event. For KF-based estimation, the equations are given as the following:

$$\overline{\mathbf{x}}_A = \mathbf{F}\left[\left(\mathbf{P}_F \cup \mathbf{P}_G\right)\left(\mathbf{P}_F^{-1}\hat{\mathbf{x}}_F + \mathbf{P}_G^{-1}\hat{\mathbf{x}}_G\right)\right] + \mathbf{Bu} + \mathbf{Gv} \tag{70}$$

$$\overline{\mathbf{P}}_A = \left(\mathbf{P}_F \cup \mathbf{P}_G\right)_A = \mathbf{F}\left[\left(\mathbf{FP}_F\mathbf{F}' + \mathbf{GQG}'\right)^{-1} + \left(\mathbf{FP}_G\mathbf{F}' + \mathbf{GQG}'\right)^{-1}\right]^{-1}\mathbf{F}' + \mathbf{GQG}' \tag{71}$$

<u>Valuation of Multiple Common Priors with Different Time References</u>

The results of the above analysis can be used with minor alterations to assess the proper valuation of common priors that have different reference frames. For the graph shown in Figure 75, the common prior sets for I_A are I_G and I_J. As with the previous

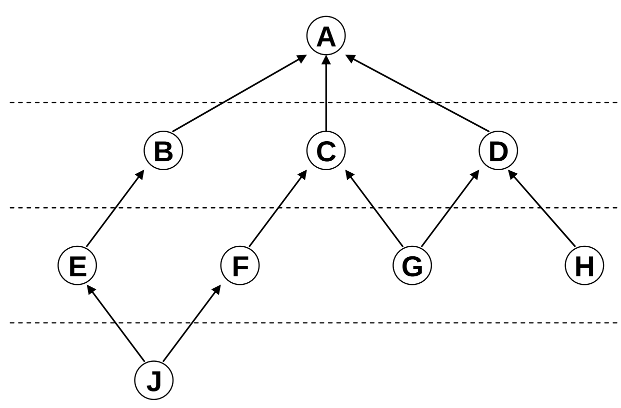

Figure 75: Network of Information Sets with Multiple Common Priors at Different Reference Frames.

example, I_A is given as the union of I_B, I_C, and I_D. Similarly, I_B and I_D are independent while I_C is composed entirely by the set of common information.

In this scenario, the following two valuation options are considered:

$$\begin{aligned}
\left[I_G \cup I_F(J)\right]_A &= \sum_{k=0}^{M} a_k \left[\sum_{j=0}^{M} a_j I_G^j \cup \sum_{j=0}^{M} a_i \left(\sum_{i=0}^{M} a_i I_J^i \right)^j \right]^k \\
I_A(G) \cup I_A(J) &= \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_F^j \right)^k \cup \sum_{k=0}^{M} a_k \left[\sum_{j=0}^{M} a_i \left(\sum_{i=0}^{M} a_i I_J^i \right)^j \right]^k
\end{aligned} \tag{72}$$

The first formulation translates the value of I_J to the same reference frame as I_G, fuses the two sets, and then translates them to the I_A reference frame. The second formulation translates the two prior sets to the I_A reference frame and then fuses them.

As with the previous analysis, the fusion at I_A is given as:

$$\begin{aligned}
I_A &= I_A(B) + I_A(D) + I_A(C) - \left[I_A(B) \cap I_A(C) + I_A(D) \cap I_A(C)\right] \\
&= I_A(B) + I_A(D) + I_A(C) - \left[I_A(G) + I_A(J)\right] \\
&= I_A(B) + I_A(D) + I_A(C) - \left[I_A(C)\right] \\
&= I_A(B) + I_A(D)
\end{aligned} \tag{73}$$

Thus once again the value of the common prior data to be removed is equivalent to value of $I_A(C)$ and is given as:

$$\begin{aligned}
I_A(C) &= \sum_{k=0}^{M} a_k \left(I_C(G) \cup I_C(J)\right)^k = \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_G^j \cup \sum_{j=0}^{M} a_j I_F^j(J) \right)^k \\
&= \sum_{k=0}^{M} a_k \left(\sum_{j=0}^{M} a_j I_G^j \cup \sum_{j=0}^{M} a_j \left(\sum_{i=0}^{M} a_i I_J^i \right)^j \right)^k
\end{aligned} \tag{74}$$

Noting that Equation (74) is equivalent to the first formulation from (72), the method for valuating the multiple common prior sets is given as:

$$[I_G \cup I_F(J)]_A = \sum_{k=0}^{M} a_k \left[\sum_{j=0}^{M} a_j I_G^j \cup \sum_{j=0}^{M} a_i \left(\sum_{i=0}^{M} a_i I_J^i \right)^j \right]^k \tag{75}$$

Namely, the valuation of multiple common prior sets with different reference frames is found by the following three steps:

1) Translate the priors to a common reference frame which is defined by the most "recent" information set;
2) Fuse the common priors;
3) Translate the fused set of priors to the current reference frame for removal from the fusion set.

For KF-based estimation, the equations are given as the following by the two formulations:

$$\bar{\mathbf{x}}_A = \mathbf{F}\left[(\mathbf{P}_F \cup \mathbf{P}_G)\left(\mathbf{P}_G^{-1}\hat{\mathbf{x}}_G + \mathbf{P}_F^{-1}(\mathbf{F}\hat{\mathbf{x}}_J + \mathbf{Bu} + \mathbf{Gv})\right)\right] + \mathbf{Bu} + \mathbf{Gv} \tag{76}$$

$$\begin{aligned} \overline{\mathbf{P}}_A &= (\mathbf{P}_G \cup \mathbf{P}_F)_A \\ &= \mathbf{F}\left[(\mathbf{FP}_G\mathbf{F}' + \mathbf{GQG}')^{-1} + \left(\mathbf{F}(\mathbf{FP}_J\mathbf{F}' + \mathbf{GQG}')\mathbf{F}' + \mathbf{GQG}'\right)^{-1}\right]^{-1}\mathbf{F}' + \mathbf{GQG}' \end{aligned} \tag{77}$$

<u>Summary of Common Priors Valuation</u>

A formal reference upon which to base the techniques used for valuating common prior sets was not found in literature. This appendix developed the valuation methods used in the Local Fusion Graph derivation and in the computer-based model. Three fundamental cases were investigated that cover the range of common prior configurations.

1) Single Common Prior: Valuation is found through the recursive application of the transition function. This function is given as the prediction function for estimation in linear dynamic systems.
2) Multiple Common Priors with Same Reference Frame: The common prior sets are first fused at their originating reference then recursively updated to the current reference frame.
3) Multiple Common Priors with Different Reference Frames: The common prior sets are translated to the reference frame of the most recent set and then fused before being recursively updated to the current fusion reference frame.

While formal proofs are not used, fundamental mathematical operations are applied to networks of information sets in a manner that demonstrates the correctness of the solutions.

LIST OF VARIABLES AND SYMBOLS

a_k	Generalized coefficient of order k
$\tilde{e}_{RMS}$	Average root-mean-square (RMS) error
$\mathbf{F}(k)$	State transition matrix
$\mathbf{G}(k)$	State control matrix
$\mathbf{H}(k)$	Measurement (observation) matrix
I	Generalized information set
i, j, m	Index variables
$\mathbf{K}(k)$	Filter gain
N	Node quantity
$\mathrm{O}(k)$	Function growth of order k
$\mathbf{P}(k)$	Covariance matrix, fused estimate
$\overline{\mathbf{P}}(k)$	Covariance matrix, common prior estimate
$\mathbf{P}(k_\mathrm{i}\mid k_\mathrm{j})$	Covariance matrix, time updated from time k_j to time k_i
$\mathbf{P}(k_\mathrm{i}\mid k_\mathrm{i})$	Covariance matrix, local estimate at time k_i
$\mathbf{Q}(k)$	Covariance of the state (process) noise
$\mathbf{R}(k)$	Covariance of the measurement (observation) noise
k_i	Time reference for time k_i
$\mathbf{u}(k)$	State input (or forcing) vector
$\mathbf{v}(k)$	State (process) noise vector
$\mathbf{w}(k)$	Measurement (observation) noise vector
$\mathbf{x}(k)$	State vector
$\hat{\mathbf{x}}(k)$	State vector, fused estimate
$\overline{\mathbf{x}}(k)$	State vector, common prior estimate
$\hat{\mathbf{x}}(k_i \mid k_j)$	State vector estimate, time updated from time k_j to time k_i
$\hat{\mathbf{x}}(k_i \mid k_i)$	State vector, local estimate
$\mathbf{z}(k)$	State measurement vector
$\beta(x;n,\mu)$	Binomial distribution with n possible states and average probability of μ
$\Phi(x)$	Probability of x
μ	Mean (average) probability

LIST OF ACRONYMS AND ABBREVIATIONS

C^3I	Command, Control, Communications and Intelligence
COMSEC	Communications security
DARPA	Defense Advanced Research Projects Agency
DoD	Department of Defense
DSO	Defense Spectrum Office (DoD)
DTN	Disruption Tolerant Networking
FAA	Federal Aviation Administration
FCS	Future Combat System
KF	Kalman Filter
IETF	Internet Engineering Task Force
IP	Internet Protocol
MANET	Mobile Ad Hoc Network
MC	Monte Carlo
MDR	Message Delivery Rate
msec	Millisecond
msg	Message
NASA	National Aeronautics and Space Administration
NCW	Network Centric Warfare
NETSEC	Network security
NSF	National Science Foundation
NeTS	Networking Technology and Systems
OSD	Office of the Secretary of Defense (DoD)
ProWiN	Programmable Wireless Networking
RF	Radio frequency
RMS	Root-mean square
s	Second(s)
SUO SAS	Small Unit Operations Situational Awareness System
TRANSEC	Transmission security
US	United States (of America)
XG	neXt Generation (Communications Program)

LIST OF REFERENCES

List of References

[1] U.S. Department of Defense. Joint Chiefs of Staff. Joint Vision 2020. Washington, DC, June 2000.

[2] U.S. Department of the Navy. Headquarters, U.S. Marine Corps. A Concept for Distributed Operations. Washington, DC, 25 April 2005.

[3] U.S. Department of Defense. Office of Force Transformation. The Implementation of Network-Centric Warfare. Washington, DC, January 2005.

[4] U.S. Department of the Navy. Headquarters, U.S. Marine Corps. Expeditionary Maneuver Warfare: Marine Corps Capstone Concept. Washington, DC, 10 November 2001.

[5] U.S. Department of Defense. Office of Force Transformation. Elements of Defense Transformation. Washington, DC, October 2004.

[6] Alberts, David S., John J. Garstka, and Frederick P. Stein. Network Centric Warfare: Developing and Leveraging Information Superiority. 2nd Ed. Washington, DC, DoD C4ISR Cooperative Research Program (CCRP), August 1999; Revised Printing, February 2000.

[7] Proakis, John G. Digital Communications. 4th Ed. McGraw Hill, 2000.

[8] Proakis, John G., and Masoud Salehi. Communication Systems Engineering. 2nd Ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[9] Wozencraft, John M., and Irwin M. Jacobs. Principles of Communication Engineering. John Wiley & Sons, 1968. reprint, Prospect Heights, IL: Waveland Press, Inc., 1990.

[10] Poisel, Richard A. Modern Communications Jamming Principles and Techniques. Norwood, MA: Artech House, 2004.

[11] Leon-Garcia, A. and I. Widjaja, Communication Networks, 2nd Ed., McGraw Hill, 2000.

[12] U.S. Department of Defense. Joint Chiefs of Staff. Doctrine for Joint Urban Operations. Joint Publication 3-06. Washington, DC, 16 September 2002.

[13] U.S. Department of Defense. Defense Advanced Research Projects Agency (DARPA), neXt Generation Communications Program Working Group. XG Vision Request for Comments. Version 2.0. Washington, DC, (Prepared by BBN Technologies, Cambridge, MA), January 2004.

[14] Alonistioti, N., A. Glentis, F. Foukalas, A. Kaloxylos. "RMP: Reconfiguration Management Plane for the Support of Policy-Based Network Reconfiguration." *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2004 (PIMRC 2004). Volume 2*. 5-8 September 2004, pp. 1394-1397.

[15] Fall, Kevin. "A Delay-Tolerant Network Architecture for Challenged Internets." *Association for Computing Machinery (ACM) Special Interest Group on Data Communications Annual Conference, 2003 (ACM SIGCOMM 2003).* Karlsruhe, Germany., 25-29 August 2003.

[16] Brown, Barbara, Paul Morris, and Craig Thompson. "SUO Communicator: Agent-Based Support for Small Unit Operations," *International Conference on Integration of Knowledge Intensive Multi-Agent Systems.* 30 Sept.-4 Oct. 2003, pp 77-82.

[17] Caprioni, Daniel, and Anthony Russo. "Small Unit Operations Situational Awareness System (SUO-SAS) Radio Architecture and System Field Testing Results." *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 1*. 13-16 Oct. 2003, pp 198-203.

[18] Jacobson, Evans, Tracy Lee, Charles Popeck, Marc Martorana, and Stan Sokolowski. "SUO/SAS Geolocation: Land Navigation Using Multiple Integrated Sensors." *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 1*. 13-16 Oct. 2003, pp. 186-191.

[19] Li, Chris, C.J. Yoon, J. Visvadar, and Paul Kolodzy. "SUO SAS Radio Intra-Networking Architecture." *IEEE Military Communications Conference, 2001 (MILCOM 2001) Volume 1*. 21-38 Oct. 2001, pp. 235-239.

[20] Liu, Y.J., Chris Li, and Christine Cruz. "Performance Optimization for a Mobile Small-Unit-Operation Situational Awareness (SUO-SAS) Radio Network." *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 1*. 13-16 Oct. 2003, pp 192-197.

[21] Randall, Alfred, and Robert C. Walter. "Overview of the Small Unit Operations Situational Awareness System." *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 1*. 13-16 Oct. 2003, pp. 169-173.

[22] Williams, Larry J. "Technology Advances from Small Unit Operations Situation Awareness System Development." IEEE Personal Communications, February 2001. pp 30-33.

[23] Williams, Larry J. "Small Unit Operations Situation Awareness System (SUO SAS): An Overview." *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 1*. 13-16 Oct. 2003, pp 174-178.

[24] Corson, S. and J. Macker, "*Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*," IETF RFC 2501, January 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2501.txt

[25] U.S. Department of Defense. Program Manager, Unit of Action. FCS 18+1+1 White Paper. Washington DC, 15 October 2004.

[26] Brown, Wayne W. et. al. "Future Combat System – Scalable Mobile Network Demonstration Performance and Validation Results," *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 2*. 13-16 Oct. 2003, pp 1286-1291.

[27] U.S. Department of Defense. Defense Advanced Research Projects Agency (DARPA), Advanced Technologies Office (ATO). BAA 05-50: Wireless Network after Next (WNaN) Proposer Information Pamphlet (PIP). Washington, DC, (Internet URL: http://fs2.fbo.gov/EPSData/ODA/Synopses/4965/BAA05-50/WNAN,PIP,29Sep05.doc), 29 September 2005.

[28] Hsu, J., S. Bhatia, M. Takai, R. Bagrodia, and M.J. Acriche. **"**Performance of Mobile Ad Hoc Networking Routing Protocols in Realistic Scenarios." *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 2*. 13-16 Oct. 2003, pp. 1268-1273.

[29] Stevens, J.; M. Feeney, L. Tipton, J. Nilsson, M. Skold, T. Ericson. "Scenario Based Analysis of Dynamic TDMA Ad-Hoc Tactical Battlefield Networking." *IEEE Military Communications Conference, 2003 (MILCOM 2003) Volume 2*. 13-16 October 2003, pp. 1332-1336

[30] Kadar, Ivan. "Issues and Challenges in Distributed Sensor Fusion: Connection-Resource Management Aspects" in "Panel Discussion on Challenges in Higher Level Fusion: Unsolved, Difficult, and Misunderstood Problems/Approaches in Levels 2-4 Fusion Research." *Proceedings of the Seventh International*

Conference on Information Fusion, 2004 (FUSION 2004) Volume 1. 2004, pp. 537-539.

[31] Beard, James K. "Tutorial AM4: Maritime Situational Awareness." *Proceedings of the Eighth International Conference on Information Fusion, 2005 (FUSION 2005).* 2005.

[32] Bar-Shalom, Yaakov, and Xiao-Rong Li. Multitarget-Multisensor Tracking: Principles and Techniques. Storrs, CT, YBS Publishing, 1995.

[33] Manyika, James, and Hugh Durrant-Whyte. Data Fusion and Sensor Management: A Decentralized Information Theoretic Approach. Ellis Horwood, 1994.

[34] Mutambara, Arthur G.O. Decentralized Estimation and Control for Multisensor Systems. CRC Press, 1998.

[35] Bar-Shalom, Yaakov, Xiao-Rong Li, and Thiagalingam Kirubarajan. Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software. John Wiley & Sons, 2001.

[36] Chong, Chee-Yee, S. Mori, and K. C. Chang, "Distributed Multitarget Multisensor Tracking." in Multitarget-Multisensor Tracking: Advanced Applications, Chapter 8, pp.247-295, ed. Y. Bar-Shalom, Artech House, 1990.

[37] Liggins, Martin E., Chee-Yee Chong, Ivan Kadar, Mark G. Alford, Vincent Vannicola, and Stelios Thomopoulos. "Distributed Fusion Architectures and Algorithms for Target Tracking." *Proceedings of the IEEE, Volume 85, No. 1,* January 1997, pp. 95-107.

[38] Chang, Kuo-Chu, Chee-Yee Chong, and Yaakov Bar-Shalom. "Distributed Estimation in Distributed Sensor Networks." in Multitarget-Multisensor Tracking: Advanced Applications, pp 23-71. ed. S. G. Tzafestas and K. Watanabe, Marcel Dekker, 1992.

[39] Chong, Chee-Yee, and Shozo Mori. "Graphical Models for Nonlinear Distributed Estimation." *Proceedings of the Seventh International Conference on Information Fusion, 2004 (FUSION 2004) Volume 1.* 28 June – 1 July, 2004, pp 614-621.

[40] Mori, Shozo, William H. Barker, Chee-Yee Chong, and Kuo-Chu Chang. "Track Association and Track Fusion with Non-Deterministic Target Dynamics." *Proceedings of the Second International Conference on Information Fusion, 1999 (FUSION'99). 6-8 July 1999.*

[41] Mori, Shozo. “Random Sets in Data Fusion: Formalism to New Algorithms.” *Proceedings of the Third International Conference on Information Fusion, 2000 (FUSION 2000) Volume 1.* 10-13 July 2000, pp TuC4-24 – TuC4-31.

[42] Stark, Henry, and John W. Woods. Probability and Random Processes with Applications to Signal Processing. 3rd Ed. Upper Saddle River, NJ: Prentice Hall, 2002.

[43] Hsu, Hwei. Probability, Random Variables, and Random Processes. McGraw Hill, 1996.

[44] McLaughlin, Samuel P., Robin J. Evans, and Vikram Krishnamurthy. “Data Incest Removal in a Survivable Estimation Fusion Architecture.” *Proceedings of the Sixth International Conference on Information Fusion, 2003 (FUSION 2003) Volume 1.* 2003, pp. 229-236.

[45] McLaughlin, Samuel, Vikram Krishamurthy, and Subhash Challa. “Managing Data Incest in a Distributed Sensor Network.” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003 (ICASSP '03), Volume 5.* 2003, pp 269-272.

[46] Hashemipour, H. R., S. Roy, and A. J. Laub, “Decentralized Structures for Parallel Kalman Filtering.” *IEEE Transactions on Automatic Control Volume 33, Issue 1*. January 1988 pp 88-94

[47] McLaughlin, Samuel, Vikram Krishnamurthy, and Robin J. Evans. “Bayesian Network Model for Data Incest in a distributed Sensor Network.” *Proceedings of the Seventh International Conference on Information Fusion, 2004 (FUSION 2004) Volume 1.* 28 June – 1 July, 2004.

[48] Biswas, Sanjit, and Robert Morris. “ExOR: Opportunistic Multi-Hop Routing for Wireless Networks,” *Association for Computing Machinery (ACM) Special Interest Group on Data Communications Annual Conference, 2005 (ACM SIGCOMM 2005).* Philadephia, PA, August 2005.

[49] Martin, Todd and Kuo-Chu Chang, “A Distributed Data Fusion Approach for Mobile Ad-Hoc Networks.” *Proceedings of the Eight International Conference on Information Fusion, 2005 (FUSION 2005).* 25-29 July 2005.

[50] Van Trees, Harry L. Detection, Estimation, and Modulation Theory. Part I, Detection, Estimation, and Linear Modulation Theory. John Wiley & Sons, Inc. 1968. reprint 2001.

[51] Holman, J. P. and W. J. Gajda Jr. Experimental Methods for Engineers. Fifth Ed. McGraw Hill, 1989.

[52] Gusfield, David. Algorithms on Strings, Trees and Sequences. Computer Science and Computational Biology. Cambridge University Press, New York, NY, USA, 1997.

[53] Pearl, Judea. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Revised 2nd Printing. San Francisco, CA: Morgan Kaufmann Publishers, Inc. 1988.

CURRICULUM VITAE

Todd W Martin received his Bachelor of Science in Aerospace Engineering from The Pennsylvania State University in 1991. He currently provides engineering and technology assessment and management services for research and development (R&D) programs as an Engineer and Program Manager at Science & Technology Associates, Inc. in Arlington, VA.

Mr. Martin has accumulated over 14 years experience in the aerospace and command, control, communications and intelligence (C^3I) fields. He has participated in numerous R&D, acquisition, and engineering analysis efforts for several US Government agencies including the National Aeronautics and Space Administration (NASA); the Federal Aviation Administration (FAA); and several organizations within the US Department of Defense (DoD) such as the Office of the Secretary of Defense Joint C^3I Decision Support Center (OSD/C^3I), the Defense Spectrum Office (DSO), and Defense Advanced Research Projects Agency (DARPA). He has also participated in the National Science Foundation (NSF) Networking Technology and Systems (NeTS) Programmable Wireless Networking (ProWiN) effort as a workshop participant and panel member.

Mr. Martin intends to continue his academic career and earn a PhD with continued studies in the areas of C^3I and artificial intelligence.