Enemy Track Based Threat Assessment in Distributed Sensing Networks

Abhay Joshi, Rajan Batta and Rakesh Nagi∗
Department of Industrial and Systems Engineering and Center for Multisource Information Fusion, 438 Bell Hall, University at Buffalo (SUNY), Buffalo, NY 14260
April 30, 2006

Abstract
Design of efficient distributed sensing/fusion networks involves issues such as sensor mobility, bandwidth capacity and reliability of communication links. Communication links between the sensors and fusion centers referred to as clusterheads are prone to jamming or foliage effects. In a recent paper, Patel et al. (2005) proposed a dynamic maximal expected coverage model for sensor networks operating under a uniform threat environment. This is useful in a situation where only the overall threat level is known. When enemy track information is available from sensors, it can be used to provide accurate assessments of threat probabilities, which vary in space and time. The first part of our work is in the calculation of these time/space dependent probabilities. We then enhance Patel et al.’s model by incorporating time dependent link failure probabilities. Computational studies involving packet level analysis are performed using OPNET network simulator, to study the benefit of using the revised threat probabilities. Results of a case study are presented, using a network centric warfare simulation testbed developed at the University at Buffalo. Both the computational and simulation results demonstrate that the track based topology management model is superior to the uniform threat topology model in terms of packets received (7-40% more), packets lost (7-50% less) and expected coverage (by 7.2%).

Keywords: Network centric warfare, military applications, wireless sensor networks, jamming, coverage.

∗Author for correspondence. E-mail: nagi@buffalo.edu
1 Introduction

Network-centric warfare (NCW), an emerging paradigm of war in the information age, is the centerpiece of the US Department of Defense’s Joint Vision 2020 approach to modern-day and future warfare. NCW generates increased combat power by networking sensors, decision makers, and shooters to achieve shared awareness, increased speed of command, high tempo of operations, greater lethality, increased survivability, and a degree of self-synchronization. In essence, it translates information advantage into combat power by effectively linking friendly forces within the battlespace, providing a much improved shared awareness of the situation, enabling more rapid and effective decision making at all levels of military operations, and thereby allowing for increased speed of execution [2]. However, providing a “common operating picture (COP)” for friendly commanders to facilitate them to have a shared understanding in decision making entails efficient distributed data fusion capability across the network. This necessitates a robust and reliable communication network topology. Architecting distributed fusion networks for NCW and their analysis is a very complex process. It requires the incorporation and evaluation of the effects of fusion processing algorithms, communication network topology algorithms, strategies for information-sharing in the network, link capacity and performance, as well as incorporating command-structure concepts of operations and inter-organizational coordination [3]. Hence, the design or analysis of fusion networks entail an empirically-based approach. This allows us to study effects and thereby determine the best approach to forming a superior composite track picture. This turn contributes to the formation of the best COP for NCW applications.

A Network Centric Warfare (NCW) simulation software testbed has been developed at University at Buffalo\(^1\) to perform empirical studies of these distributed problems at

\(^1\)Refer to \url{http://www.eng.buffalo.edu/~nagi} for more details about the project
the architectural level, accounting for dynamic network topologies, jamming of nodes, node-specific tracking algorithms and data fusion algorithms. The testbed can potentially be used for real-time battle scenarios, or for strategic planning, to predict and simulate potential movements of friend or foe. The tool allows the user to design a range of battlefield scenarios and evaluate and test algorithms to perform network topology management, sensor report simulation, target tracking, and data fusion. The testbed consists of a scenario generator, tracking and distributed fusion algorithms, dynamic network topology manager and performance evaluator and information visualization tools. The dynamic network topology manager utilizes tactical models proposed by Joshi, Batta and Nagi [4] to provide robust communication scheme that maximizes the overall efficiency of the communication system in a tactical setting. The proposed tactical models are extensions of Patel, Batta and Nagi [1]’s strategic model and suggest strategies to deploy a given number of clusterheads, over a specific time horizon, allowing relocation of clusterheads and minimizing the weighted sum of expected demand covered and relocation cost. The NCW testbed utilizes a capacitated network topology design model [4], which considers bandwidth capacity at clusterheads. The model also incorporates threat due to enemy attack (such as jamming) in design time by assigning a constant probability of failure to all the communication links during all the time periods. However, this is approximate in the sense that it either under - or over - estimates the instantaneous threat. Moreover, as the testbed incorporates tracking algorithms, some level of information about the enemy location and its intent is available. This intelligence can be used to revise the threat probabilities, so they vary in space and time. A brief news coverage about the software can be found in [5].

To further motivate the need to assess and revise the threat probabilities we cite some of the following intuitive reasons. Knowing the whereabouts of enemy targets and recognizing their identity can help gain significant information about their elec-
tronic systems. The information also enables the commander to conduct protective measures to ensure continued use of friendly electronic systems. In summary, it offers the commander the opportunity to gain tactical advantage. Hence, it is crucial for the friendly forces that keep track of the enemy movements to make improved strategic and tactical decisions. This paper aims to model the threat probability by utilizing the enemy track information. The model adopts a numerical integration based solution methodology to evaluate the link failure/threat probabilities.

The remainder of the paper is organized as follows: Section 2 presents a literature review. Section 3 describes the network topology design problem under a threat environment and suggests a methodology to revise the threat or the communication link failure probabilities using tracking information. Computational study and analysis of the results using OPNET network simulator are presented in Section 4. A case study using NCW testbed is presented in Section 5. Finally we conclude the discussion in Section 6.

2 Literature Review

Adaptive sensor networks, due to continuous adaption of sensing and communication topology, allow us to achieve efficient coverage and data fusion, even with environmental changes/threats or sensor failures (We note here that the communication engineering literature refers to these networks as ad hoc networks, however the term adaptive better reflects the military context). These networks, in military applications, may gather intelligence in battlefield conditions, track enemy troop movements, monitor a secured zone for activity, or measure damage and casualties. Due to application criticality, these networks must be resilient to individual node and link failures due to jamming, foliage effect or instrument malfunctions. Jamming is one of the widely accepted method for disrupting an adversary’s communication network. Jammers are devices capable of
obscuring or denying information. Masking jammers deny information by submerging the transmission in interference, which may consist of signals like noise. The complexity of masking jammers depends on the device being jammed (radars, communication receivers or fuzes) [7]. For example, transmission of a narrow band of noise may be adequate to make communication over the radio link unintelligible to the listener, while transmission of a wide band of noise may be necessary to deny information to a radar using a correlation process for detection. In general, the effectiveness of jamming depends on - target link distance (i.e., distance between the message transmitter and receiver), the distance between the jammer and the targeted receiver, radio line of sight distance between the jammer and the targeted receiver, antenna polarization, effective radiated power of the jammer and the message transmitter, bandwidth compatibility between the jammer and message receiver and weather, terrain and vegetation [6]. We note here that the radio line of sight distance refers to the distance that a radio wave would travel between the jammer and the message transmitter - this includes extra travel to negotiate around obstacles that may be present, like buildings, hills, trees, etc.

Several researchers have addressed the issue of communication jamming in a wireless sensor network and its effect on the performance of the network. Xu et al. [13] discuss radio interference attacks on wireless sensor networks. They study the feasibility and effectiveness of jamming attacks on wireless networks and examine the critical issue of detecting the presence of jamming attacks. They also propose four different jamming attack models that can be used by an adversary to disable the operation of a wireless network, and evaluate their effectiveness in terms of how each method affects the ability of a wireless node to send and receive packets. Karlof and Wagner [14] consider routing security in wireless sensor networks and introduce two classes of novel attacks against sensor networks - sinkholes and HELLO floods, and analyze the security
of the major sensor network routing protocols. Papadimitratos and Haas [15] present a route discovery protocol that mitigates the detrimental effects of jamming to provide correct connectivity information. Wood and Stankovic [16] studied link jamming systematically and proposed using error correcting codes to cope with the collision attack, rate limitation to deal with the exhaustion attack, and small frames to deal with an unfairness attack. Shi and Perrig [17] discuss the security issues in wireless sensor networks and consider mechanisms to achieve secure communication in these networks. Although, all these research papers and the others cited in the literature consider some type of threat (due to jamming or enemy attack) to communication protocols for wireless sensor networks, no one suggests the method of assessing the threat across a military theater. In this work we consider threat during the network topology design for wireless sensor network and propose a numerical integration based method for evaluating this threat due to the presence of enemy targets. The proposed method derives motivation from the research work in collision probability assessment of space objects (satellite, debris, etc.). Hence, we review some relevant literature in this area.

Muinonen, Virtanen and Bowell [11] introduce new techniques for the computation of the collision probability for earth-crossing asteroids in the case of short observational arcs and/or small numbers of observations. The techniques rely on the orbital element probability density computed using statistical orbital ranging. They provide upper bounds for the collision probability in the linear approximation and using the rigorous probability density. The authors optimize the statistical ranging technique for orbit determination, and establish a Monte-Carlo technique for the computation of the collision probability for short-arc earth-crossing asteroids. Alfriend et al. [8] investigate the effect of errors in the covariance on probability of collision and the sensitivity of probability of collision to the encounter geometry. They develop a method to obtain the maximum possible probability of collision for any encounter. Their results
indicate that small changes in covariance size induces rapid changes in probability of collision values. Akella and Alfriend [9] present an alternative but equivalent definition of probability of collision and develop a direct method to obtain the same. They compare the proposed method with the one suggested by Foster [12] and prove that both approaches are equivalent in the sense that they yield the same formula for the probability of collision. Chan [18] showed that it is permissible to combine the error covariance matrices of two orbiting objects to obtain a relative covariance matrix as long as they are represented in the same coordinate frame. The combined covariance matrix has an associated three-dimensional probability density function that represents the uncertainty in relative distance between the two objects.

Patel et al. [1] proposed the dynamic maximal expected covering location (DMEX-CLP) model with the objective of maximizing the expected sensor demand covered by locating a given number of clusterheads. Joshi et al. [4] builds on [1] and proposes two generalizations of this strategic model from a tactical perspective. The first generalization considers variable relocation cost and the second one considers limited bandwidth capacity at clusterheads; both [1] and [4] assume a uniform threat environment. In this paper we develop an efficient method to evaluate and revise threat or link failure probabilities by utilizing the enemy track information provided by the tracking algorithm. A numerical implementation of the computation scheme is also presented.

3 Solution Methodology

3.1 Problem Overview

We consider a battlefield scenario, land, sea and air or a combination of these, consisting of friendly sensors (soldiers, sonars on ships, submarines or unmanned aerial vehicles (UAVs)) and enemy targets (radars, submarines, UAVs). The friendly sen-
Sensors are deployed to gather information about the adversary, and the enemy targets are positioned with an intention of obstructing (jamming) the friendlies. The sensors (friendlies) act as information sources by identifying the threats to the system. The information provided by various sources is fused. We consider a decentralized data fusion environment wherein the data from a group of sensors (referred to as clusters) is processed together at their corresponding fusion centers or clusterheads. This processed data from the clusterheads is then transmitted either to the other clusterheads or sent to the command center (sink node). In the military, Airborne Warning and Control Systems (AWACS), a radar mounted tank, or the captain of an infantry unit may be clusterheads. The data transfer between the clusterheads and sensors takes place through wireless communication. The clusterheads are assumed to fuse the information gathered from the sensors and the clusterheads connected to it. The sensors do not communicate with one another, but the clusterheads can communicate with the other clusterheads. Thus, sensors and clusterheads are functionally different.

Sensors and/or clusterheads, being mobile, communicate through wireless medium. Typical issues related to wireless communication are transmission/communication range, available bandwidth, and network reliability. Successful execution of a mission is greatly influenced by reliability of the communication network. But in a battlefield scenario, a communication link between a sensor and a clusterhead is prone to enemy attack, such as jamming, and hence has an associated probability of failure. This link failure probability may also be due to instrument malfunction, foliage effect and terrain effects. However, in this work we are only concerned about link failure due to jamming. To ensure network reliability under these hostile conditions it is crucial to gain and react to knowledge about the enemy capabilities and their locations. This type of information is provided by the target tracking algorithm, which can be effectively used to design a robust and reliable communication network topology. In this work,
we concentrate on the reliability issue of sensor network topology management and present a methodology which utilizes tracking information to predict the threat due to the presence of enemy targets. The threat information provided by this model is then utilized in the strategic model proposed by Patel [19] to design a network topology scheme for the described scenario. The strategic model yields the clusterhead locations as a function of time (over a discrete set of time intervals). Once the clusterhead locations are known for a particular time period, we can easily determine the set of sensors that are covered by each clusterhead, and also determine which sensors are covered multiple times.

To develop the threat assessment model, along with the assumptions stated in [1], we make the following additional set of assumptions relating to sensors, clusterhead and target behaviors.

1. A sensor is jammed with some probability if,
   (a) it lies within a known jamming radius of an enemy target’s jamming device’s positional uncertainty region (refer to Figure (1)) and
   (b) the enemy target’s jamming device is in active mode

2. A target can jam the sensor with some probability if,
   (a) it has jamming device mounted on it
   (b) its jamming device is set to active mode and
   (c) the sensor lies within its known jamming radius

3. Location of the sensors are known at any given instance (since they belong to the friendly force). We note that sensors are allowed to move but we assume knowledge sensor location at each discrete time instant.
4. Potential clusterheads are selected from a set of discrete locations. Typically, the cardinality of the feasible locations is much greater than the number of clusterheads to be selected.

5. Each link fails independently of all other links, i.e., a link failure is independent of the other link failures.

![Figure 1: Snapshot representation](image)

The sensor has communication/coverage radius $R$ and sensing range $R'$, generally $R > R'$. It senses objects within its range and sends this information to the connected clusterhead for fusion. It utilizes various tracking algorithms such as the Kalman filter, Particle filter, etc., to estimate the position of the target(s). For our analysis we assume that the fused target information is available and is provided either by a sensor or a clusterhead utilizing a Kalman based filtering algorithm. The discrete-time Kalman filter assumes that the underlying system is dynamic linear system. It also assumes that the posterior density at every time step is Gaussian and hence exactly and completely characterized by two parameters, its mean and covariance [21]. In other words, the measurement noise and process noise are Gaussian, white, and independent.
of each other. The algorithm has two distinct phases: Predict and Update. The predict phase uses the estimate from the previous timestep in a dynamic model to produce an estimate of the current state. In the update phase measurement information from the current timestep is used to refine this prediction to arrive at a new, more accurate estimate.

A constant velocity Kalman filter model assumes that the true state of the target at time $t$ is evolved from the state at $(t - 1)$ according to,

$$x_t = F_t x_{t-1} + B_t u_t + w_t,$$

(1)

where,

$F_t$ is the state transition matrix which is applied to the previous state $x_{t-1}$

$B_t$ is the control-input matrix which is applied to the control vector $u_t$

$w_t$ is the process noise which is assumed to be drawn from a zero mean multivariate normal distribution with covariance $Q_t$, $w_t \sim N(0, Q_t)$

At time $t$ an observation (or measurement) $z_t$ of the true state $x_t$ is made according to

$$z_t = H_t x_t + v_t,$$

(2)

where, $H_t$ is the observation model which maps the true state space into the observed space and $v_t$ is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $R_t$, $v_t \sim N(0, R_t)$. The initial state, and the noise vectors at each step $\{x_0, w_1, ..., w_t, v_1, ..., v_t\}$ are all assumed to be mutually independent.

The conditional distributions, that are calculated in a Kalman filter algorithm are as follows:

$$p(x(t - 1) \mid Z(t - 1)) = \mathcal{N}(x(t - 1); \hat{x}(t - 1| t - 1), P(t - 1| t - 1)),$$

(3)

$$p(x(t) \mid Z(t - 1)) = \mathcal{N}(x(t); \hat{x}(t| t - 1), P(t| t - 1)),$$

(4)

$$p(x(t) \mid Z(t)) = \mathcal{N}(x(t); \hat{x}(t| t), P(t| t)),$$

(5)
where, $\mathcal{N}(x; m, P)$ is a Gaussian density with argument $x$, mean $m$ and covariance $P$, i.e.

$$
\mathcal{N}(x; m, P) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{\det(P)}} \exp\left(-\frac{1}{2}(x - m)^T P^{-1} (x - m)\right).
$$

(6)

The terms $(x - m)$ and $P$ represent the state and covariance estimate of the target conditioned on the measurements up to time $t$. Here the state vector is defined as:

$$
x = [x, \dot{x}, y, \dot{y}, z, \dot{z}]_k^T.
$$

where, $x$, $y$, $z$ are the positions and $\dot{x}$, $\dot{y}$, $\dot{z}$ are velocities. Now, as stated in the assumptions, a communication link between a sensor and a clusterhead is assumed to fail if this particular sensor is jammed by the target. So in order to determine the probability that the communication link between sensor-clusterhead pair fails, we need to estimate the probability that this sensor is jammed. A sensor is assumed to be jammed with some probability if it lies within the jamming radius of target’s jamming device set to active mode of transmission. As stated in the assumptions, this also requires knowledge of the exact characteristics, such as type of jamming device mounted, its mode of operation (active or passive), jamming radius and transmission frequencies of the target. In reality it is difficult to get such information. Therefore, instead of finding the probability that a sensor lies within the target’s jamming range, we find the probability that the target is within the communication/coverage radius of the sensor. This can be determined using positional information of the sensor and target.

Consider a sensor $S$ and a target $T$ moving in a 3-dimensional Cartesian space. Let $S(t)$ be the position vector of the sensor $S$ at time $t$. Again, we note that sensors can move but we assume knowledge of their positions at each discrete time instance - thus sensor mobility is being modeled. The position vector is defined as:

$$
S(t) = [x_S(t), \ y_S(t), \ z_S(t)].
$$

(7)
Let $T(t)$ be a real-valued random vector denoting state of the target $T$ at time $t$. The state vector is defined as:

$$X(t) = [x_T(t), \ x'_T(t), \ y_T(t), \ y'_T(t), \ z_T(t), \ z'_T(t)]^T,$$  \hspace{1cm} (8)

where, $x_T(t), y_T(t), z_T(t)$ are the positions in the $x$, $y$ and $z$ directions, $x'_T(t), y'_T(t), z'_T(t)$ are the velocities in the $x$, $y$ and $z$ directions, respectively. $T$ denotes matrix transpose.

Let $d(S(t), T(t))$ be a distance function between sensor $S$ and target $T$ at time $t$. We define a new random variable $d$ as:

$$d = \sqrt{(x_S - x)^2 + (y_S - y)^2 + (z_S - z)^2}.$$

Let $E$ be an event such that $d \leq R$, i.e. the distance between the sensor and target is less than or equal to the coverage radius of the sensor, where $R$ is the coverage radius of sensor $S$. To find the probability of this event we need to find the statistics $F_d(d)$, which are,

$$F_d(d) = P(d \leq R) = P(d(S(t), T(t)) \leq R) = P(T(t) \in R) = \int \int \int_{x,y,z \in R} f(x, y, z) \, dx \, dy \, dz,$$  \hspace{1cm} (9)

where,

- $R$ is the domain representing the region where the inequality $d \leq R$ is satisfied, and

- $f(x, y, z) = \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{\det(P)}} \exp(-\frac{1}{2}(x - m)^T P^{-1}(x - m)).$
Substituting for \( f(x, y, z) \) we obtain,

\[
p_{st} = F_d(d) = \int \int \int_{x,y,z \in R} \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{\det(P)}} \exp(-\frac{1}{2}(x - m)^T P^{-1}(x - m)) \, dx \, dy \, dz
\]

\[\text{(10)}\]

\[
= \int_0^R \int_{\sqrt{(R^2-y^2)}}^{\sqrt{(R^2-x^2-y^2)}} \int_{\sqrt{(R^2-x^2-y^2)}}^{(2\Pi)^{\frac{3}{2}} \sqrt{\det(P)}} \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{\det(P)}} \exp(-\frac{1}{2}(x - m)^T P^{-1}(x - m)) dx \, dy \, dz.
\]

\[\text{(11)}\]

Here, as stated earlier, \((x - m)\) and \(P\) represents the state estimate and covariance estimate of the target. Transforming (11) to spherical co-ordinates we obtain,

\[
p_{st} = \int_0^R \int_0^{2\Pi} \int_0^\Pi \frac{1}{(2\pi)^{\frac{3}{2}} \sqrt{\det(P)}} \exp(-\frac{1}{2}(x - m)^T P^{-1}(x - m)) d\theta \, d\phi \, dR \, R^2 \sin(\phi)
\]

\[\text{(12)}\]

where,

\[x = R \sin(\phi) \cos(\theta) \quad y = R \sin(\phi) \sin(\theta) \quad z = R \cos(\phi)\]

Solving equation (12) numerically, we obtain the probability that the target \( T \) lies within the coverage radius \( R \) of the sensor \( S \) at time instant \( t \). This is, as per model assumptions, the probability that sensor \( S \) is jammed by target \( T \) during time \( t \). However, we have \( n \) number of targets/tracks present in the scenario, each based on its characteristics, capable of jamming the sensor with some probability. We note that many tracks are in fact “false” tracks, in that they are generated by data fusion but do not represent actual targets. Hence the \( n \) refers not to the actual number of targets but to the potential number of targets. Therefore, we need to find the jamming probability of sensor \( S \) due to all the individual tragets/tracks. Let \( p_{stT_1}, p_{stT_2}, p_{stT_3}, \ldots, p_{stT_n} \) be these individual jamming probabilities, i.e., probability of sensor \( S \) being jammed by
target $T$ at time $t$ where, $T = 1, \ldots, n$. Here each outcome has its own probability of occurring, which implies that jammers are operating independently. Therefore the combined jamming probability of sensor $s$ at time $t$ is given by,

$$p_{st} = 1 - [(1 - p_{stT_1})(1 - p_{stT_2})(1 - p_{stT_3}) \ldots (1 - p_{stT_n})].$$

(13)

The general framework of the methodology is shown in Figure 2. The jamming probability, calculated using (13) represents the probability of link failure between a sensor and the clusterhead. This link failure probability represents a more realistic picture of the threat for the sensor in the theater and hence can be used to design reliable communication network protocol. We use this link failure probability in the DMEXCLP model proposed by Patel et al. [1] in an anticipation of capturing the present situation to yield better results. Replacing $p$ with $p_{st}$ in the objective function of DMEXCLP model given in Appendix(A), we obtain:

(P) Maximize

$$\sum_{t=0}^{T} \sum_{s \in \Theta} \sum_{j=1}^{n} (1 - p_{st})p_{st}^{j-1}d_{st}y_{jst} - \sum_{t=1}^{T} \sum_{i \in \Delta} Cw_{it},$$

subject to: Constraints (14) to (20).

The proposed methodology is implemented in the NCW simulator to test and analyze its benefit in the network topology design model suggested by [1]. The details of the implementation and subsequent packet level analysis comparing the track based network topology design model with the uniform threat based topology design model in presented in the next two sections (Sections (4) and (5)).
4 Computational Results

In this section we perform packet level analysis using the OPNET 11.0 network simulator [20] to demonstrate the benefit of using revised threat probabilities. We first generate a set of scenarios using the scenario generator tool (SceneGen) of NCW simulator. The scenario generator utilizes the scenario parameters given in Table 1 to generate the problem instances. Each of these scenarios are then simulated, under two different modes - i) non-track based or uniform threat mode ($P$ model), ii) track based threat mode ($P_{kt}$ model), for the stipulated duration of time to obtain optimal topology schemes. The topology scheme provided by the network topology design model [4] contains information about the clusterhead candidates chosen and the assignment of sensors to the these chosen clusterheads at any point in time. In addition to this, relevant information such as node trajectories, sensor demands, link failure probabilities associated with sensors per time period and multiple coverage information (i.e., secondary clusterheads covering the sensors) is also provided. The multiple coverage

Figure 2: Flowchart for the track based threat model ($P_{kt}$)
information is used to redirect the sensor data or packet to secondary clusterheads in case of failure of the primary clusterhead. We utilize this set of data to perform a packet level analysis in OPNET.

As a first step we construct the network model corresponding to each scenario in OPNET. The network model specifies entities/nodes (sensors and clusterheads) as well as their physical locations, interconnections and configurations. Initial network topology is created by placing the sensors and clusterheads randomly in the workspace. The area selected for simulation is 108x108 km$^2$ in size. The entities in the network model move within the region according to predefined trajectories. A snapshot of the OPNET network model is shown in the Figure (3). Here the triangular objects represent clusterheads with communication rings (coverage radius) and circular objects represent sensors.

<table>
<thead>
<tr>
<th>Table 1: Model parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario Parameter</td>
</tr>
<tr>
<td>Size of area (km$^2$)</td>
</tr>
<tr>
<td>Sensor/Clusterhead Velocity Range (km/min)</td>
</tr>
<tr>
<td>Target Velocity Range (km/min)</td>
</tr>
<tr>
<td>Sensor Sensing Range (km)</td>
</tr>
<tr>
<td>Model Parameter</td>
</tr>
<tr>
<td>Relocation cost ($C$)</td>
</tr>
<tr>
<td>Max. # of Clusterheads that can be chosen ($n$)</td>
</tr>
<tr>
<td># of potential Clusterheads ($\Delta$)</td>
</tr>
<tr>
<td># of sensors ($\Theta$)</td>
</tr>
<tr>
<td>Length of the time horizon ($T$)</td>
</tr>
<tr>
<td>Demand data range for sensors ($d$)</td>
</tr>
<tr>
<td>Coverage radius, km ($U$)</td>
</tr>
<tr>
<td>Clusterhead bandwidth capacity ($Q$)</td>
</tr>
</tbody>
</table>

A sensor node has a processor and a transmitter, whereas a clusterhead node has a processor and a receiver both with fixed radio range. The communication between a sensor and a clusterhead is through wireless radio medium. The properties such as transmission rates, frequencies, bandwidth, power levels of transmitters and receivers
are specified using real life sensor specifications [22] data and are set as follows: (1) Receiver/transmitter channel: Data rate - 10.71 Mbps, Bandwidth - 400-800Mhz, Minimum frequency - 14.40 GHz, (2) Transmitter channel - Power: 2.0 W. Since the clusterheads are assumed to be identical, their data packet processing times are considered to be same. The nodes in the network model behave according to the actions specified in their respective process models (coded using C++). During the initialization stage of the network simulation run, all nodes read the trajectory file generated for that specific scenario and change their positions accordingly. In addition, the sensor reads the assignment (to the respective clusterheads), demand, and link failure probability information. Based on this data the sensor sends the packets to its assigned clusterhead through wireless radio channel at predefined intervals using periodic transmission strategy (decided by its scan rate of the sensor). We assume a unicast communication model (i.e., packets are received only by a specific clusterhead) for our simulation. The packet thus received at the clusterhead is either accepted or rejected based on its
validity.

We also incorporate threat or link failure aspect into the network model by assigning failure probabilities to the transmission links between the sensors and the clusterheads. To implement this we first read time varying link failure probabilities \( p_{kt} \) of sensors, provided by the threat model. Then during every instant of packet transmission stage of a sensor, we generate a random number between 0 and 1 and compare it with the \( p_{kt} \) value of this particular sensor at that time. If the \( p_{kt} \) value is less than the generated random number then the transmission/communication of packet to the destination clusterhead is assumed to fail. Otherwise the packet is successfully delivered. With this general framework and set of specifications we run the simulation for the stipulated duration of time \( T \) and collect the following statistics for comparison.

- Total number of packets sent by the sensors to their clusterheads
- Total number of packets received by the chosen clusterheads
- Total number of packets lost due to link failure between the nodes
- Total number of packets lost due to collision of the packets sent to the same receiver channel of a clusterhead (i.e., network traffic flow/congestion)

Here note that, for fair comparison, the topology schemes provided by both \( \mathbf{P} \) and \( \mathbf{P}_{kt} \) models are simulated under the same threat environment.

The results of the packet level analysis are presented in Table 2. The results indicate that the number of packets received in the \( \mathbf{P} \) model is less than the \( \mathbf{P}_{kt} \) model in all the problem instances. This includes two effects of better topology design in response to link failure as well as packet collision due to congestion. Overall the percentage of packets received in the \( \mathbf{P} \) model is 7-40% less than the \( \mathbf{P}_{kt} \) model. The last column reflects the number of packets lost due to link failure only. This is higher for the \( \mathbf{P} \)
Table 2: Model Comparison Results - Track based (Pkt) and Non-track based (P) threat model

<table>
<thead>
<tr>
<th>No</th>
<th># of packets sent</th>
<th># of packets received</th>
<th># of packets lost due to link failure</th>
<th># of packets lost due to collision</th>
<th>% Increase in packets received P vs Pkt</th>
<th>% Decrease in packets lost P vs Pkt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>88</td>
<td>92</td>
<td>31</td>
<td>37</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>214</td>
<td>232</td>
<td>90</td>
<td>106</td>
<td>14</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>81</td>
<td>82</td>
<td>14</td>
<td>19</td>
<td>51</td>
<td>45</td>
</tr>
<tr>
<td>4</td>
<td>112</td>
<td>117</td>
<td>78</td>
<td>84</td>
<td>31</td>
<td>25</td>
</tr>
<tr>
<td>5</td>
<td>69</td>
<td>75</td>
<td>47</td>
<td>57</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>61</td>
<td>63</td>
<td>37</td>
<td>45</td>
<td>21</td>
<td>15</td>
</tr>
<tr>
<td>7</td>
<td>81</td>
<td>85</td>
<td>53</td>
<td>61</td>
<td>25</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>193</td>
<td>202</td>
<td>117</td>
<td>131</td>
<td>65</td>
<td>56</td>
</tr>
<tr>
<td>9</td>
<td>93</td>
<td>97</td>
<td>60</td>
<td>67</td>
<td>27</td>
<td>19</td>
</tr>
<tr>
<td>10</td>
<td>107</td>
<td>117</td>
<td>69</td>
<td>78</td>
<td>33</td>
<td>24</td>
</tr>
</tbody>
</table>

model compared to the Pkt model and the percentage difference ranges from 7-50%. Thus the performance of topology scheme of Pkt model much superior compared to P model in terms of percentage of packets lost and packets received. It was also observed that the number of packets lost due to collision is higher in the Pkt model than the P model. This is because the number of packets sent by the sensors in the Pkt model is 1-10% higher (due to better coverage) than the P model, resulting in more collisions.

Figure 4: Effect on varying coverage radius on number of packets sent and received - Track based threat model (Pkt)

We also conduct some tests to study the effect of varying the model parameters such
as bandwidth capacity and coverage radius on the performance measures. The results indicate that increase in clusterhead bandwidth capacity ($Q$) increases the packet loss due to collision as more sensors are assigned to the same clusterhead. It was also observed that the number of packets received at the clusterhead increase, since the clusterhead can cover more sensor demand when its bandwidth is increased. The coverage radius of the clusterhead has a similar effect on the performance measures. The packet loss due to collision and number of packets sent increase with increase in coverage radius (refer to Figure (5)). But number of packets received increases with the increase in coverage radius up to a certain value then it decreases due to the packet loss resulting from collisions (refer to Figure (4)).

## 5 Case Study

We present a case study using the NCW simulator to illustrate how the proposed threat revision model can be used in network topology design of sensor networks in the battlefield environment. For our case study, we consider a scenario with 5 clusterheads (air), 9 sensors (air) and 7 enemy targets (air). The scenario is generated using the scenario generator tool in NCW simulation testbed. The scheme provides the user with
flexibility of choosing various networking algorithms - column generation, MOBIC [23], relocation and no-relocation heuristics [4]. Additionally, the testbed allows the user to execute the topology management scheme under two modes - uniform or non-track based threat mode ($P$) and track based threat mode ($P_{kt}$). The track based threat evaluation option utilizes the model proposed in this paper to estimate the threat or link failure probabilities. The uniform threat option do not revise the link failure probabilities and uses the average value of the time varying threat. Thus, we first simulate the scenario under track based threat mode and determine the average threat in the theater and then simulate it under uniform threat option.

The track based threat revision method in the NCW testbed employs the following scheme to revise the link failure probabilities. During the initial stages of the simulation, due to the unavailability of the information about the targets (i.e. tracks), the link failure probability values of all the sensors/clusterhead links are set to a value deemed appropriate for the theater of operation (perhaps $p$). As time progresses the simulation produces tracks, providing state estimate and covariance estimate of the targets, from the tracking algorithms. The threat revision method takes the sensor positions and target positions and covariances as inputs and revises (calculates) the link failure probabilities using methodology explained in Section 3 (refer to the flowchart in Figure (2)). The networking algorithm utilizes these updated link failure probabilities to determine the appropriate network topology for that time period.

To consider a case simulation, we consider Buffalo east region map of 108 $km^2$ area. The topography is generated using the terrain data (digital elevation model format) and vector map data (digital line graph format). The entities (sensors, clusterheads and targets) are randomly located in the theater which travel with velocities in the uniform range of 10-20 km/min. The other model parameters chosen for the study are as follows. The coverage radius ($U$) is 25 km, constant relocation cost range is 5-10,
target jamming radius is 10 km, clusterhead bandwidth capacity is 50 Mbps, maximum number of clusterhead to be chosen is 5, and sensor demand range is 5-10 Mbps. The waypoints for all the entities are generated for every $1/10^{th}$ of a second. Therefore, the total number of time periods, $T$ under consideration is 2000. However, as it impractical to simulate for the entire time period under consideration (due to the complexity of the problem), we divide the entire time period into a sequence of smaller horizons and solve the network calculation problem over their periods. Moreover, to emulate an infinite horizon and to capture and utilize the events from the previous time periods we adopt a rolling horizon approach and overlap the intervals in the sequence. The length of the overlap and time interval ($T$) are specified by the user and in our case it is 30 time steps (3 seconds) and 15 time steps (1.5 seconds) respectively. However, this method requires the network calculation model to be invoked several times (134 invokes in this case).

With these modifications to the model, we simulate the scenario for the stipulated simulation duration of 200 seconds. The snapshots of the simulation results at different time intervals is shown in Figures (6), (7), (8), (9), (10), and (11). Figure (8) shows the chosen clusterheads (the large transparent circles representing coverage radius) and the sensors assigned to them during the first 3 seconds of the simulation. The dotted line represents the communication link between a clusterhead and a sensor. Figure (9) shows inter-clusterhead communication while Figures (10) and (11) display the target tracks (represented by ellipsoids) and topology scheme respectively. In addition to the clusterhead and sensor level information, the model also provides the user with various performance measures (link failure probabilities of sensors, % of expected coverage during each time period and over the interval, number of single, double and triple coverage, etc.) at the global level. These measures to can be used to evaluate the effectiveness of various networking algorithms (Greedy, CG [4]) and compare them
with the other algorithms (MOBIC [23] and Geographical based). In this case we compare the performance of the track based threat model ($P_{kt}$) with the uniform threat model ($P$). The results of the simulation indicate that the $P_{kt}$ model performs better than $P$ model in terms of expected coverage value by 7.2%. In other words the $P$ model underestimates the expected coverage by 7.2%. Figure (12) shows how threat or link failure probabilities of sensors evolve with time. The results suggest increase in threat probabilities with time for all the sensor (1 to 9). This is due to the lack of information about the target locations during the initial stages of the simulation.

We also conduct parametric analysis to find out the effect of varying the target jamming radius, sensor sensing range and the number of targets on the threat probability. The results indicate that the increase in jamming radius of a target increases the threat probabilities of the sensors within its vicinity. Similarly, increase in the sensor sensing range of a sensor increases its threat probability value as it can sense more number of targets within its neighborhood (i.e., provide more accurate picture of the threat in the theater). Finally, increasing the number of targets in the theater increases the threat probabilities of the sensors (within their range) as the newly added targets, apart from the existing ones, have the chance of jamming the sensors within their range.

6 Conclusion

In this work we proposed a numerical integration based method to determine time dependent threat or communication link failure probabilities between the sensors and the clusterheads due to electronic jamming. The model utilizes the tracking information (state and covariance estimates) provided by the tracking algorithms to calculate the link failure probabilities. We then improve the network topology design model proposed by Patel et al. [1] by incorporating the calculated link failure probabilities. The performance of this track based network topology design model ($P_{kt}$) is compared
Figure 6: Snapshot of the simulation - initialization phase

Figure 7: Snapshot of the simulation showing platform characteristics
Figure 8: Snapshot of the simulation showing sensor to clusterhead communication during first 3 seconds

Figure 9: Snapshot of the simulation showing inter-clusterhead communication during first 3 seconds
Figure 10: Snapshot of the simulation showing target tracks

Figure 11: Snapshot of the simulation showing topology scheme during 60 to 90 seconds time interval
with the uniform threat based topology design model ($P$) by performing packet level analysis using the OPNET network simulator. The results of the analysis indicate that the $P_{kt}$ model outperforms $P$ model in terms of percentages of packets received and packets lost (lesser # of packets lost in $P_{kt}$ model). Parametric analysis to study the behavior number of packets lost due to collision, number of packets sent and received with respect to change in model parameters such as coverage radius and bandwidth capacity are also presented. Finally, a case study using NCW simulator is presented to demonstrate how the proposed track based threat determination model can be utilized in communication network topology design to provide “common operating picture (COP)” for friendly commanders and help them to have a shared understanding in decision making. The results of the case study indicate that $P_{kt}$ model provides better expected coverage than the $P$ model.

**Acknowledgment**

This research was performed under the sponsorship of Rosettex Technology and Ventures Group. The authors gratefully acknowledge their support.
Appendix A

Parameters:

\( \Delta = \) set of potential clusterhead locations.
\( \Theta = \) set of sensors.
\( n = \) maximum number of clusterheads to be chosen.
\( T = \) maximum number of time periods in the horizon under consideration.
\( C = \) cost per unit change in the number of clusterheads at any location \( i \) (one-half of relocation cost).
\( U = \) the distance beyond which a sensor is considered “uncovered”.
\( D_{ikt} = \) distance between potential clusterhead location \( i \) and demand node \( k \) at time \( t \).
\( d_k = \) demand per period of node \( k \).
\( p = \) probability of a link failure per period (between any facility and demand node). \((0 < p < 1)\)
\( r_{ikt} = \begin{cases} 
1, & \text{if } D_{ikt} < U; \\
0, & \text{otherwise.}
\end{cases} \)

Decision Variables

\( x_{it} = \begin{cases} 
1, & \text{if clusterhead } i \text{ is chosen at time } t, \\
0, & \text{otherwise.}
\end{cases} \)

\( y_{jkt} = \begin{cases} 
1, & \text{if sensor } k \text{ is covered by at least } j \text{ clusterheads at time } t, \\
0, & \text{otherwise.}
\end{cases} \)

\( w_{it} = \) positive difference in the number of clusterheads at location \( i \) between time \( t - 1 \) and time \( t \).
DMEXCLP Model [1]

(P) Maximize \( \sum_{t=0}^{T} \sum_{k \in \Theta} \sum_{j=1}^{n} (1-p)p^{j-1}d_{kyjkt} - \sum_{t=1}^{T} \sum_{i \in \Delta} Cw_{it} \),

subject to:

\[
\sum_{j=1}^{n} y_{jkt} - \sum_{i \in \Delta} R_{ikt} x_{it} \leq 0 \quad \forall \quad k \in \Theta, t = 0, \ldots, T, \quad (14)
\]

\[
\sum_{i \in \Delta} x_{it} \leq n \quad \forall \quad t = 0, \ldots, T, \quad (15)
\]

\[
w_{it} \geq x_{it-1} - x_{it} \quad \forall \quad i \in \Delta, t = 1, \ldots, T, \quad (16)
\]

\[
w_{it} \geq x_{it} - x_{it-1} \quad \forall \quad i \in \Delta, t = 1, \ldots, T, \quad (17)
\]

\[
x_{it} \in \{0, 1\} \quad \forall \quad i \in \Delta, t = 0, \ldots, T, \quad (18)
\]

\[
w_{it} \geq 0 \quad \forall \quad i \in \Delta, t = 1, \ldots, T, \quad (19)
\]

\[
0 \leq y_{jkt} \leq 1 \quad \forall \quad j = 1, \ldots, n, k \in \Theta, t = 0, \ldots, T. \quad (20)
\]
References


